



JAPANESE PATENT OFFICE

# 4

## PATENT ABSTRACTS OF JAPAN

(11)Publication number: 07044477

(43)Date of publication of application: 14.02.1995

OK

(51)Int.Cl.

G06F 13/00  
G06F 3/00  
G06F 3/14  
H04L 12/28  
H04N 5/00  
H04N 5/445  
H04N 5/765  
H04N 5/907  
// G06F 9/44  
G06F 12/00

(21)Application number: 05189636

(22)Date of filing: 30.07.1993

(71)Applicant:

(72)Inventor:

CANON INC

TAKAHASHI FUMIAKI

MAMIYA AKIRA

SUGA AKIRA

KAWAMURA HIDEAKI

AIZAWA TAKASHI

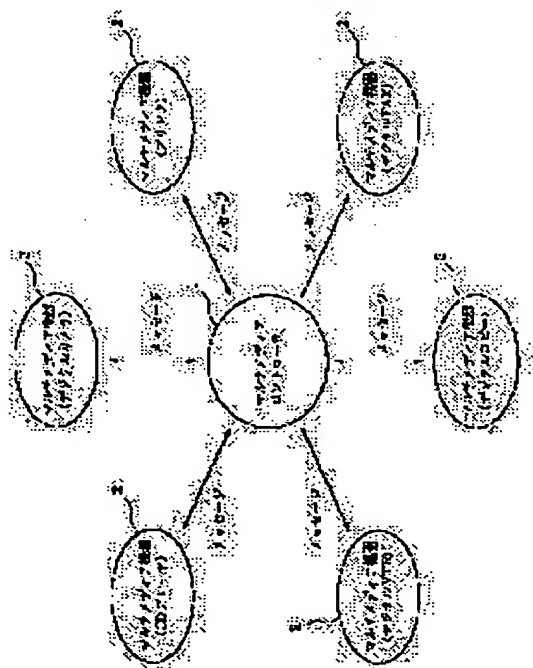
HATORI KENJI

### (54) CONTROL SYSTEM FOR MULTI-MEDIUM EQUIPMENT

(57)Abstract:

**PURPOSE:** To provide the environment utilizing a multi-medium equipment in common from other controller in a transparent way via a LAN without use of a special software in the multi-medium equipment.

**CONSTITUTION:** In the system in which plural multi-medium equipments 2 and a controller 1 to control them are connected in a network and the plural multi-medium equipments 2 and the controller 1 are capable of sending receiving a message and data based on an object orientation via the network, a display means and a pointing means are provided to the controller 1 and a pattern symbolizing each of the plural multi-medium equipments 2 is displayed on the display means and the user uses the pointing means to set a link between the patterns to designate input output relation of data between the multi-medium equipments.



---

**LEGAL STATUS**

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

特開平 7-44477

(43)公開日 平成7年(1995)12月14日

(51)Int. Cl. <sup>6</sup> G 06 F 13/00 3/00 3/14	識別記号 3 5 5 A 3 7 0 A	庁内整理番号 7368-5 B A	FI	技術表示箇所
				H 0 4 L 11/00 3 1 0 D H 0 4 N 5/762 K OL (全 5 5 頁) 最終頁に続く
(21)出願番号 特願平 5-189636	(71)出願人 000001007 キャノン株式会社 東京都大田区下丸子3丁目30番2号 株式会社内			
(22)出願日 平成5年(1993)7月30日	(72)発明者 高橋 安明 東京都大田区下丸子3丁目30番2号キャノン株式会社内 (72)発明者 岡宮 明 東京都大田区下丸子3丁目30番2号キャノン株式会社内 (72)発明者 菅 章 東京都大田区下丸子3丁目30番2号キャノン株式会社内 (74)代理人 井理士 丸島 徹一			

(54)【発明の名称】 マルチメディア機器の制御システム

(57)【要約】

【目的】 マルチメディア機器において、特別なソフトウェアを必要とせず、またLANを介して他のコンピュータから自動的に共通的にマルチメディア機器を利用できる環境を提供することにある。

【構成】 複数のマルチメディア機器及びこれらを制御するための制御装置がネットワーク上に接続され、該ネットワークを介して前記複数のマルチメディア機器及び制御装置が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムにおいて、前記制御装置には表示手段及びボイシング手段が付随し、前記表示手段により、前記複数のマルチメディア機器それぞれを象徴する図柄が表示され、利用者が該ボイシング手段により、これらの図柄の間にリンクを張ることによりマルチメディア機器間のデータの出入力関係を指定するように構成したマルチメディア機器の制御システム。

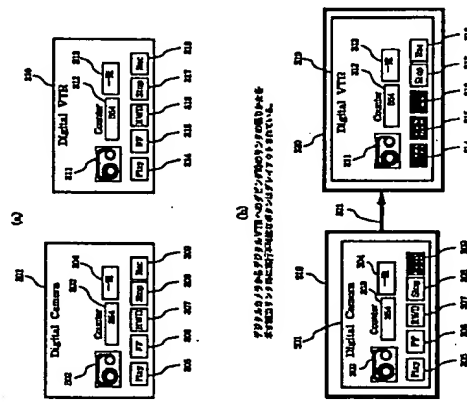


図1 (a) (b) (c) マルチメディア機器の制御システム

(2) 【特許請求の範囲】

【請求項1】 複数のマルチメディア機器及びこれらを制御するための制御装置がネットワーク上に接続され、該ネットワークを介して前記複数のマルチメディア機器及び制御装置が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムにおいて、前記制御装置には表示手段及びボイシング手段が付随し、前記表示手段により、前記複数のマルチメディア機器それぞれを象徴する図柄が表示され、利用者が該ボイシング手段により、これらの図柄の間にリンクを張ることによりマルチメディア機器間のデータの出入力関係を指定することを特徴としたマルチメディア機器の制御システム。

【請求項2】 請求項1において、前記ネットワーク上に接続されているマルチメディア機器を象徴する図柄およびリンクを表示するためのユーザインターフェース及びその表示手段を備えたことを特徴とする備えたことを特徴とするマルチメディア機器の制御システム。

【請求項3】 複数のマルチメディア機器及びこれらを制御するための制御装置がネットワーク上に接続され、該ネットワークを介して、該複数のマルチメディア機器及び制御装置が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムにおいて、前記複数のマルチメディア機器を制御するための制御装置に、利用者のデータ入力出力関係指定手段、データフォーマットの整合性判断手段を備え、利用者が複数のマルチメディア機器間のデータ入力出力関係を指定するデータフォーマットの整合性を判断するように構成したことを特徴とするマルチメディア機器の制御システム。

【請求項4】 請求項4において、前記整合性判断手段により、データフォーマットの整合性が得られない時に、データフォーマットの整合性が得られないためにデータの送受信が不可能であることを利用者に通知する手段を備えたことを特徴とするマルチメディア機器の制御システム。

【請求項5】 複数のマルチメディア機器及びこれらを制御するための制御装置がネットワーク上に接続され、該ネットワークを介して、該複数のマルチメディア機器及び制御装置が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムを構成するマルチメディア機器において、前記制御装置から送信される出力または入力可能なファイルフォーマットに関する問い合わせに対する返答手段を備えるとともに、返答する際に複数のデータフォーマットが入力または出力可能であるときにこれらデータフォーマットの優先順位を前記制御装置に通知する手段を備えたことを特徴とするマルチメディア機器。

【請求項6】 複数のマルチメディア機器及びこれらを制御するための制御装置がネットワーク上に接続され、

前記ネットワークを介して、該複数のマルチメディア機器及び制御装置が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムにおける該複数のマルチメディア機器を制御するための制御装置において、オブジェクト指向に基づくクラスが記述されたデータを外部より導入する手段を有し、更に、該クラスよりオブジェクトを生成する手段を有することを特徴とするマルチメディア機器の制御システム。

【請求項7】 複数の機器がネットワークを介して互いにデータを送受信できるシステムにおいて、該系列データがリアルタイムで送受信されてきた時に、データ記録モードでない時には、該系列データのヘッダ部分の情報を一時的に記憶する手段を有し、データ記録モードとなった時に該ヘッダ部分を記憶手段より読み出し、該ヘッダ部分を該系列データの先頭に配置して記録媒体へて記憶する手段を備えたことを特徴とするデータ記録装置。

【請求項8】 複数の機器がネットワークを介して互いにデータを送受信できるシステムにおいて、該ネットワークを介してプログラムデータが送信されてきたことを判定する判定手段と、前記判定手段により前記プログラムデータの受信が判定された場合に前記プログラムデータに基づき自動的にそのプログラムを起動する起動手段とを備えたことを特徴とする制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、文字・音声・静止画・動画等の各種情報を取り扱うマルチメディア機器のシステム制御に用いて好適なものである。

【0002】

【従来の技術】 従来アナログ技術を中心としていたオーディオ・ビデオ・TV等のAV機器においては、近年急速にデジタル化が進んできている。また、文字・静止画情報のデジタル化の普及と合わせて、いわゆるマルチメディアとして文字・音声・静止画・動画情報がコンピュータの中で統一的に取り扱われるようになってきた。

【0003】

【発明が解決しようとしている課題】 しかしながら、現在マルチメディア機器（デジタルカメラ、CD-ROMプレーヤ、スキャナ、サウンドボード、ビデオボード等の音声出力機器、映像出力機器等）をコンピュータで利用する場合は、デバイスドライバというソフトウェアをコンピュータにインストールしなければならない。新しいマルチメディア機器に対しては新たなアプリケーションソフトまたはデバイスドライバをコンピュータにインストール（Operating System）ごとに用意しなければならない。そのため、ソフトウェアの開発負担が大きく、効率的かつ高価な制御が不可能であるという問題があった。

汎用コンピュータ上に専用OSとアプリケーションソフトウェアを搭載して実行する事もできる。

【0024】次に図2に、マルチメディアコンピュータとマルチメディア機器の双方向性通信路の成立の為の物理的接続形態を(a)～(c)に示す。

【0025】図(a)は、SBSバス(ANSI X3.131-1986)で採用しているデジチエイン接続方式を、図(b)はEthernet(IEEE 802.3) 10baseTで採用されているスター型接続方式を、図(c)はEthernet 10base2/5で採用されている直列型接続方式をそれぞれ示すものである。

【0026】また接続形態では、GPIB(IEEE 488)の様な(a)～(c)の混合方式や、Ethernetでも(b)、(c)の混合方式がある。また通信方式でも、光ケーブルやISDNを用いる方式など、図2以外にも様々な組み合わせや選択が出来る。

【0027】本発明では、この双方向性の通信路の成立方法やどれを選択するかは特に言及しない。但し、通信方式の違いによる物理的制限(転送速度・接続台数・延長・コネクタ形状等)は、メッセージの相互通信に、はプロトコルの階層が違いために問題にはならないが、周辺機器の相互接続を確保に行うためには、最低限一種の物理的(メカ的・電気的)に共通なインターフェースを持つ必要がある。

【0028】また動画像のような高速度なデータ通信を実現する為には、Ethernetより高速なFiber Distributed Data Interface)やP-ISDN等の光通信を用いる方法が考えられるが、ここでは説明の為に、簡便で普及しているEthernet 10base2(T)を共通の通信コネクタとして持っているものとして説明を進めることにする。

【0029】次に一般的なマルチメディア機器のハードウェア面での内部ブロック図を図3に示す。

【0030】複数のマルチメディア機器はそれぞれ4のLANを介して、コントローラと接続されている。今MはEthernetであるので、その通信プロトコル(TCP/IP)を処理するインターフェース部20が設けられている。これは専用LSI等の利用で実現出来る。ここでは送られてきたメッセージそのものが取り出されたり、逆にコントローラへメッセージが送りだされる。メッセージの例として、Objectiveでは、一般形は、以下で表現される。

【0031】[対象オブジェクト メソッド名:引数] 他の言語では表現が異なるが、基本的には同様で、以下の指定が行われる。

【0032】(1) 対象オブジェクトの指定  
(2) メソッド(実行させる処理)の指定  
(3) 引数(パラメータ)があれば、その指定  
このメッセージの取り扱いは図3のソフトウェアの流れで説明する。

【0033】マルチメディア機器の内部には、10の内部バスを介して、すべてのソフトウェア処理・ハードウ

られ、このオブジェクトIDに基づいてデータの送受信を行うことにより、論理的にデータの送先が決定される。オーディオビデオ機器間のデータ送受信関係を構築・修正する際に物理的接続をつなぎかえる必要はなく、データ送受信関係の構築・修正にかかわるオブジェクトの内部データを変更するのみでよいことになる。

【0019】また各オブジェクトは、コントローラに管理される為、自身が持っている機能・コントロール手段をコントローラに送りだす機能を有している。これにより、今までの様に予めコントローラ側に制御プログラムを準備する必要がなくなり、単にコントローラと接続するだけで制御を実現出来る様にした。コントローラは、接続されているオブジェクトから送られた上記制御手段を、実際に制御を指示する人間に提示・操作させる為の手段を有する。このことにより、一つのインターフェース上でネットワークに接続されている全てのオーディオビデオ機器を操作することが可能となる。さらにコントローラ内に機器間の接続を管理する手段を設け、この管理手段が上記表示手段を用いて機器間の接続状況を表示することにより、多数の接続関係が構築されているも、利用者が接続状況を容易に認識することが出来る。

【0020】また、上記操作手段により、利用者は接続の編集作業を行うことができ、この内容を前記管理手段にメッセージングすることにより前記管理手段は利用者の意図を解釈し、各機器のデータ入出力に関する情報を保持するオブジェクトに対して、データ入出力に関する情報を問い合わせ、機器間のデータの整合性を判別し、表示手段にこの判別結果を表示することにより、利用者は機器間のデータ送受信可/不可を容易に知ることが出来る。

【0021】また上記表示手段により、各オーディオビデオ機器を象徴するオブジェクトが表示され、利用者がこれらの表示されているオブジェクト同士を上記操作手段により結線して接続関係を構築することによって、利用者による過った接続も起こりえない。

【0022】図1は、本発明のオブジェクト指向の概念を取り入れた、マルチメディアコンピュータとマルチメディア機器の論理的な接続形態を示す。1のマルチメディアコンピュータを中心に、2の各マルチメディア機器はそれぞれ1対1で各種情報の直接対話が行えるように通信路が確立されている。その通信路を介してメッセージを相互に通じる事により制御を行うものである。マルチメディア機器は、具体的にはCDプレーヤー・デジタルVTR・デジタルカメラ・デジタルTV等のAV機器やデジタルFAX・デジタルコピー機・プリンター等のOA機器など、すべてのマルチメディアデータを取り扱う機器を対象としている。

【0023】またコントローラは、ここでは専用の機器を想定しているが、パソコンやワードプロセッサWSの

ら、このオブジェクトIDに基づいてデータの送受信を行うことにより、論理的にデータの送先が決定される。オーディオビデオ機器間のデータ送受信関係を構築・修正する際に物理的接続をつなぎかえる必要はなく、データ送受信関係の構築・修正にかかわるオブジェクトの内部データを変更するのみでよいことになる。

【0019】また各オブジェクトは、コントローラに管理される為、自身が持っている機能・コントロール手段をコントローラに送りだす機能を有している。これにより、今までの様に予めコントローラ側に制御プログラムを準備する必要がなくなり、単にコントローラと接続するだけで制御を実現することが出来る。

【0015】またコントローラは、接続されているオブジェクトから送られた上記制御手段を、実際に制御を指示する人間に提示・操作させる為の手段を有しており、これによつてコントローラは集中的にマルチメディア機器を管理出来る様になり、また新しいマルチメディア機器に対して新たな準備をすることなく対応出来るという柔軟性・拡張性を実現することが出来る。

【0016】尚、本発明に用いられるオブジェクト指向の概念自体は、例えば、『石塚:オブジェクト指向プログラミング、アスキー出版、1988。』、『酒井:オブジェクト指向入門、オーム社、1990。』、『G.J.コックス:オブジェクト指向のプログラミング、トッパン、1988。』等の参考文献に詳しく説明されているため、以下の本発明に実施例の説明においては、基本的な技術説明は省略する。

【0017】このオブジェクト指向は、近年のプログラミング環境の効率化といった観点で注目を集めているが、更にOSやマルチメディアデータベースにも広く活用することができ、特にオブジェクト指向で特徴的な概念は、

(1) カプセル化  
(2) 継承  
(3) メッセージング

【0018】そして本発明ではオーディオビデオ機器及び、コントローラをネットワーク上に接続し、それらオーディオビデオ機器を、個々にオブジェクトとしてとらえ、コントローラはそれらのオブジェクトを統一的に管理するという手法を用いている。オーディオビデオ機器はネットワーク上に接続されるとオブジェクト指向に基づいたメッセージ及びデータを送信する際

に送信の宛て先として用いられるオブジェクトIDが与え

【0005】またこの方法では、一般的にはLANに接続された他のコンピュータから、そのマルチメディア機器を物理的に使用することが出来ないため、LANを介して各コンピュータから各周辺機器にアクセスできるようなマルチメディアシステムのコンセプトを実現することができないものであった。

【0006】本発明の課題は、上述の問題点を解決することにある。これらのマルチメディア機器において、上記アプリケーションソフトウェアやデバイスドライバ等の特別なソフトウェアを必要とせず、またLANを介して他のコントローラから透過的に共通的にマルチメディア機器を利用できる環境を提供し、特にマルチメディア機器間でデータの送受信を行う際に利用者に煩為なユーザーインターフェースをもつてマルチメディア機器間のデータ送受信関係を構築できるようにし、実際のデータ送受信の際には、コントローラを介さずに、マルチメディア機器間のみでデータ送受信を行うことを目的とする。

【0007】課題を解決するための手段 本発明は上述の問題点を解決するためになされたもので、その特徴とするところは、複数のマルチメディア機器及びこれらを制御するための制御機器がネットワーク上に接続され、該ネットワークを介して前記複数のマルチメディア機器及び制御機器が互いにオブジェクト指向に基づいたメッセージ及びデータの送受信が可能なシステムにおいて、前記制御機器には表示手段及びポインティング手段が備付し、前記表示手段により、前記複数のマルチメディア機器それぞれを象徴する図柄が表示され、利用者が該ポインティング手段により、これらの図柄の間にリンクを張ることによりマルチメディア機器間のデータの出入力関係を指定するように構成したマルチメディア機器の制御システムにある。

【0008】作用 これによつて、オーディオビデオ機器間のデータ送受信関係を構築・修正する際に物理的接続をつなぎかえる必要はなく、データ送受信関係の構築・修正にかかわるオブジェクトの内部データを変更するのみで実現することが可能となる。

【0009】また各オブジェクトは、予めコントローラ側に制御プログラムを準備する必要がなくなり、単にコントローラと接続するだけで制御を実現出来る様にした。

【0010】またコントローラは、接続されているオブジェクトから送られた上記制御手段を、実際に制御を指示する人間に提示・操作させることにより、一つのインターフェース上でネットワークに接続されている全てのオーディオビデオ機器を操作することが可能となる。

【0011】またコントローラ内に機器間の接続を管理する手段を設け、この管理手段が上記表示手段を用いて

ニア制御を行う110CPUと、プログラムや初期値や固有情報が格納された12のROMと、一時データやレジスタ情報等の内部パラメータを格納したり一時データの受け渡しに際してワーク領域として使われる13のRAMと、内部媒体あるいは外部媒体に格納されている15のマルチメディアデータをアクセスする14のデータI/Oと、17のモーター等の駆動部分とを制御する16のメカ系駆動部分と、19のディスプレイSWやLED等の表示系電気駆動部分を制御する18の電気系制御部分がある。また、15のマルチメディアデータは、画像・音声・文字等のデジタルデータが格納される部分であるが、CD-ROM・HD-RAM等の光ディスクやDDC、DAT等の磁気テープ媒体あるいは半導体メモリカードなど他の形態があり得る。

【0034】次にマルチメディアコントロールのハードウェア面での内部ブロック図を図4に示す。図面において4のLANを介して、マルチメディア機器と接続されている。今LANはethernetであるので、その通信プロトコル(TCP/IP)を処理するインターフェース部31が設けられている。これは専用LSIを用いて実現出来る。ここで送られてきたメディアセッセージそのものが取り出されたり、逆にマルチメディア機器へメディアセッセージが送りだされる。

【0035】マルチメディアコントロールローラの内部には、3.0の内部バスを介して、すべてのソフトウェア処理・ハードウェア制御を行う21のCPUと、プログラムや初期値や固有情報を持つ22のROMと、一時記憶域や機器状態等の内部パラメータを格納したリプログラムの実行に際してワーク領域として使われる23のRAMがある。25のマルチメディアファアイリング装置は、内部媒体あるいは外部媒体のいずれによりマルチメディアデータの格納・検索・再生・編集を行う。そのアクセスコントロールの条件のうち24のデータ1/0で、29のスイッチSWやLED等の表示系の電気部分と制御する28の電気駆動部と、マンマシンインターフェースを構成する27のディスプレイと、その表示制御を行う26のディスプレイコントローラと、図示しないマウス等のポインティングデバイスがある。

【0036】図6は、マルチメディア機器のソフトウェア面でのシステム構成図を示す。図3で示した内部ブロック図が57のハードウェアにあたる。これらのハードウェアを制御するものの基本制御部を行うのが58のOSである。OS自体は特に限定されないが、リアルタイム性と同時に複数のプログラムを並行して実行するマルチタスクの機能を持つ合わせていることが望ましい。このOSの上に、マルチメディア機器のオブジェクト化を実現するために、マルチメディア機器毎に固有のクラスライブラリ59を持っている。

【0037】また図示していないがコントローラから制御されるための、自身コントロールパネルやコントローラに関するライブラリを持っていて、これをコントローラと稼働時に送信することにより、マルチメディア機

器固有の制御をコントローラ側から実現させる。またタ  
イマーや算術演算を行うC関数60がある。

【0038】最上位の階層には、マルチメディア機器本体のコントロールと、マルチメディアコントロールとしての通信やユーザーインターフェースを受け持つUIのアプリケーションソフトウェアがある。このアプリケーションによって、マルチメディア機器本体が一つのオブジェクトとしてコントロールからメッセージのやり取りで種々の制御や実行を行なうことができる。また内部パラメータはインスタンス変数として隣が足しや減算が行える。

【0039】図5は、マルチメディアコンテンツローラのソフトウェア面でのシステム階層を示す。図4で示した内部ブロック図が50のハードウェアにわたる。これらのハードウェアを制御する為の基本的制御を行うのが51のOSである。こゝでも50自体は特に限定されないが、リアルタイム性とマルチタスクの機能を持つてゐることが望ましい。

【0040】このOSの上に、搭載された複数のマルチメディア機器のコントローラ画面の表示や全体のシステム稼働状態の表示や印刷及びデータ入出力の切り替えなどのGUI (Graphical Users Interface) 全般を行なう52のWindow Server がある。53の共通クラスライブラリーは、あらかじめコントローラ側で用意している。ボタン、スライダポリューム、テキスト表示エリア等のユーザーインターフェースやコントロールに関する基本的で共通的な部品群 (オブジェクト群) が格納されている。

【0041】 図に55の固有クラスライブラリは、格納されているマルチメディア機器固有のパネル表示やコントロールに関する部品群（オブジェクト群）が格納されている。この固有ライブラリは先に説明したように、マルチメディア機器がシステムに接続される毎にその機器から送られてきて増加していく。これらの具体的な手順は後述する。またタイマーや算術演算を行うC関数554がある。最上位の階層には、格納されているマルチメディア機器全体のコントロールと、マルチメディア機器との通信やユーザーインターフェースを受け持つ556のアプリケーションソフトウェアがある。

【0042】このコントローラとマルチメディア機器間の具体的な制御の流れとメッセージのやり取りについてこれから説明を行う。

【0043】図7はマルチメディア機器をマルチメディア  
アクセントローラに接続する前の状態を示す図である。図  
7において4はデジタルデータとの通信を行うためのLAN  
N、1はシステム全体の動作を制御するマルチメディア  
コントロールである。2はLAN4に接続されるマルチ  
メディア機器の構造を一般化したものである。205は  
マルチメディア機器のアクセントローラに搭載し、システム全体  
の管理を行うソフトウェア・オブジェクト（以後オブジェ  
クトと略す）であるシステム・オブジェクト・タワー・オブジェクト  
と略す）である。

である。

【0044】1064はLAN4上の他のオブジェクトによってオブジェクト化されたマルチメディア機器として機能するオブジェクトである。マルチメディア機器オブジェクト1064はさらに3つのオブジェクト1065、1066、1067から構成されている。

【0045】1065はマルチメディア機器2の大部分の機能を現示するためにハーウェアの制御を行うマルチメディア機器コントローラオブジェクト、1066は他の機器からデジタルデータのLAN4を介した入力を受け持つマルチメディア機器データ入力オブジェクト、1067は他の機器へのデジタルデータのLAN4を介した出力を受け持つマルチメディア機器データ出力オブジェクトである。

【0046】1061はマルチメディア機器2をマルチメディアコントローラ1にLAN4を介して接続した際に、マルチメディアコントローラ1内に生成されるマルチメディア機器代理オブジェクトの仕様を記述するマルチメディア機器代理オブジェクト1062をファイルである。マルチメディア機器代理オブジェクト1063はマルチメディア機器代理オブジェクト1061にマルチメディア機器2の操作パネルの仕様を記述するマルチメディア機器コントローラパネルオブジェクト1064を記述する。マルチメディア機器2のデータ入出力の代理を行うデータ入出力代理オブジェクト1065を記述する。特にマルチメディア機器コントローラパネルオブジェクト1064はマルチメディア機器2の操作をGUIで行うためのコントローラパネルを記述するGUI記述言語の機能を実現している。

【0047】図8はLANにマルチメディア機器2が接続されたときの状態を説明する図である。図8において1068はマルチメディアコントロールローラ1内に生成されるオブジェクトでありマルチメディアコントロールローラ1内においてマルチメディア機器2の代理として機能するマルチメディア機器代理オブジェクト1068である。マルチメディア機器代理オブジェクト1068はマルチメディア機器2のコントロールパネルとして機能するマルチメディア機器コントロールパネルオブジェクト1069、データ入力の際にもマルチメディア機器データ入力オブジェクト1066の代理として機能するマルチメディア機器データ入力代理オブジェクト1070、同様にマルチメディア機器データ出力オブジェクト1067の代理として機能するマルチメディア機器データ出力代理オブジェクト1071から構成される。

【0048】図9は一般的なクラスライブラリーの構成を示す図である。図9において1079は同様な性質を持ったオブジェクトに共通な性質や機能を定義しオブジェクト生成のためのテンプレートとして機能するクラス

第pクラス1085までのp個のクラスをライブラリーとしてまとめたものをクラスライブラリー1086と称し、すべてのオブジェクトは特定のクラスに所属する。1080はクラスに属するオブジェクトが持つ内部変数のデータ型と名称、データ処理手段をあらわす内部関数（一般的にクラスメソッドに相当する）のデータ型と名称を定義するクラス定義部、1081はクラスメソッドのアクセスを可能にするためにクラスメソッドの各コードへのポインターをテーブル化したクラスメソッドテーブル、1082は第1関数コード1083から第k関数コード1084までのk個のクラスメソッドの関数コードを格納するコード部である。

【0049】図10は一般的なオブジェクトの構造を示した図である。図10において234はオブジェクトであり、クラスメンバデータ部235のポインタ格納部244、メソッド部239、内部データ部235によって構成される。メンバ部239は第1データ処理手段40、第2データ処理手段241をはじめとする第mデータ処理手段242までのm個のデータ処理手段で構成されている。235は内部データ部であり、第1内部データ236、第2内部データ237をはじめとする第n内部データ238までのn個の内部データで構成されている。

【0050】内部データ部235を構成する個々の内部データは個々のオブジェクトに固有であるためオブジェクト内部に持っているが、メモリ部が持つデータ処理手段はクラスが同じであればオブジェクト間で共有できるため第1データ処理手段244から第mデータ処理手段242までのデータ処理手段はクラスメモリ部243によってクラス毎に管理され、同じクラスに属する複数のオブジェクトから共有される。クラスメモリ部243はクラスメモリ部244に格納されるポインタによって各オブジェクトから参照される。

【0051】メッセージ通信手段245は他のオブジェクトからのメッセージを受け取り処理手段246に送る。処理手段246は該メッセージを解析し該メッセージに対応するデータ処理手段をメモリ239（図解にはクラスラスタブル243から）検索し実行させる。データ処理手段はメッセージに添付されたデータ、内部データ部245に存在する内部データ、外部データに対し所定の処理を実行する。処理によっては他のオブジェクトに対しメッセージを送出すものもある。その場合該メッセージはメッセージ通信手段245を介して他のオブジェクトに送出される。

【0052】図11はシステムディレクター-オブジェクト205の構造を示す図である。同図において1072はクラスメソッドテーブルへのポインタ-格納部であり、システムディレクター-クラスラスメソッドテーブル



の構成要素の対応を説明する図である。各基本的な構成要素が所属するクラスはあらかじめクラスライブラリー1081に定義されている。図21が示すとおりコントロールVTR中に保持されており、マルチメディアコントロールVTRコントロールパネルオブジェクト22.1の各構成要素間がデジタルVTRコントロールパネルオブジェクト22.1を構成するオブジェクトとして機能する。

【0072】図21においてコントロールパネル表示面231のフレームはパネルクラスのVTRコントロールパネルオブジェクト284 (ID=1) に対応している。コントロールパネルの表示確保メニュー232はメニュークラスのパネルビュー一般メニューオブジェクト285 (ID=2) に対応している。タイムカウントオブジェクト286はフォーマツクラスのタイムカウントオブジェクト286 (ID=3) に対応している。巻き戻しボタン表示269はボタンクラスの巻き戻しボタンオブジェクト287 (ID=4) に対応している。再生ボタン表示270はボタンクラスの逆転再生ボタンオブジェクト288 (ID=5) に対応している。一時停止ボタン表示271はボタンクラスの一時的停止ボタンオブジェクト289 (ID=6) に対応している。再生ボタン表示272はボタンクラスの再生ボタンオブジェクト290 (ID=7) に対応している。早送りボタン表示273はボタンクラスの早送りボタンオブジェクト291 (ID=8) に対応している。停止ボタン表示274はボタンクラスの停止ボタンオブジェクト292 (ID=9)、録画ボタン表示275はボタンクラスの録画ボタンオブジェクト293 (ID=10) に対応している。

【0073】コントロールモード選択部266はボタングループクラスのコントロールモード切り替えオブジェクト294 (ID=11) に対応している。第1のスイッチボタン267はラジオボタンクラスのデフォルトボタンオブジェクト295 (ID=12) に対応している。第2のスイッチボタン268はラジオボタンクラスの上級ボタンオブジェクト296 (ID=13) に対応している。

【0074】次に図21に示したデジタルVTRコントロールパネルオブジェクト221を構成するオブジェクトのうち、例として再生ボタンオブジェクト再生ボタンオブジェクト290の生成に関して説明する。

【0075】図22は再生ボタンオブジェクト290の生成に関する説明図である。図22において297、298、299、300、601、602、603、604、605、606、607、608、609、610、611はデジタルVTR代理オブジェクト記述ファイル210のオブジェクトコントロールパネルオブジェクト記述247に記述された要素を示している。

【0076】297はオブジェクト認識情報であり、クラス名298、オブジェクトID299、所属オブジェクトID300から構成される。601は第1オブジェ

クト描画情報であり、描画位置・大きさ情報602、形状・色情報603、オブジェクト画像604から構成される。605は第2オブジェクト描画情報であり、描画位置・大きさ情報606、形状・色情報607、オブジェクト画像608から構成される。609はオブジェクトトリック情報であり、リンク先オブジェクトID610と送出メッセージ611から構成される。

【0077】290はクラスとデジタルVTR代理オブジェクト記述ファイル210のオブジェクトコントロールバリエーションオブジェクト247の情報から生成されるポインタオブジェクト再生ポインタオブジェクトである。613はクラスメモリッドテーブルのポインタ-格納部であり、ポインタクラスクラスメモリッドテーブル625をポイントするポインタ-を格納している。ポインタクラスクラスメモリッドテーブルはポインタのオブジェクトが生成される際にポインタオブジェクトの内部変数を初期化するポインタ-初期化手段626とポインタオブジェクトの表示を描画するポインタ-描画手段627、利用者がポインタオブジェクトの描画位置をマウス等のポインティングデバイスのカーソル230で指示してクリック動作を行ったときにその動作に反応してポインタオブジェクトがクリックされたことをポインタの表示-制御に繋げるなどで示すとともに他のオブジェクトにメッセージを送出するクリック反応手段から構成されている。

【0078】これらのポタンクラスメソッドテーブルが保持する名データ処理手段の定義はクラスに記載されており、再生ポタンオブジェクト290だけでなく、他のポタンクラスに属するすべてのオブジェクトから共通に参加されもちいられる。614はメッセージ通信手段、615は処理制御手段である。616はメソッド部であり、620は内部データ部である。内部データ部620は、オブジェクトID621、ポタン状態データ622、描画パラメータ623、リンクデータ624から構成されている。再生ポタンオブジェクト290だけでなくポタンクラスに属するすべてのポタンオブジェクトが持つべき内部データの型はクラスに記載されている。

【0079】システムディレクターオブジェクト205はデジタルVTR代理オブジェクト記述ファイル210を読み込んで各オブジェクトを生成するが、図22の例ではオブジェクト認識情報297のクラス名298の記

の所属オブジェクト情報をもとにシステムディレクター  
オブジェクト205はオブジェクト間の包含関係を知  
り、複数のオブジェクトから構成されるオブジェクトを  
複合オブジェクトとして生成する。

【0080】ボタン描画手段627は描画パラメータ623とボタン状態データ622にもとづいて再生素子オブジェクト290を描画する。ボタン描画手段627はボタンオブジェクト生成時と所画オブジェクトの移動時に自動的に実行される。

【0081】第1オブジェクト描画情報601は押されてない時のボタンの描画情報625を記述している。描画位置・大きさ情報603は再生ボタンオブジェクト290を描画する際のデジタルVTRコントロールパネル221における描画位置と大きさを示す。矩形情報604は記述されている。該矩形情報は押されてない時のボタンの描画情報625の(X1、Y1)、(X2、Y2)のようにデジタルVTRコントロールパネル221の座標系において該矩形情報を規定する座標情報、例として左上と右下の座標によって表現される。押されてないときの再生ボタンオブジェクトの描画は形状・色情報603もしくはオブジェクト画像604にもとづいて行われる。形状・色情報603は線の書き方色の塗り方などオブジェクトを描画するための言語で記述されている。オブジェクト画像604はビットマップデータで表現されている。一般に前者で表現したほうがデータ量は少なくてむけ後者の方が自由度が高い。

【0082】第2オブジェクト描画情報605は第1オブジェクト描画情報601と同様の方法で呼ばれたとき、ボタンの描画情報628を記述している。第1オブジェクト描画情報601と第2オブジェクト描画情報605をもとにして描画パラメータ623が決定される。リソースデータ624はオブジェクト描画情報609にもとづいて設定され、送出メッセージとして「play」が、リンク先オブジェクトIDとしてリンク先オブジェクトIDが設定されるが、メッセージを送出する際、受信側のオブジェクトがシステム全体で一様に決定される目的で、デジタルVTR203をLAN4に接続した際、システムディレクターオブジェクト205がデジタルVTRに割り当てるデバイスIDをリンク先オブジェクトIDに付加した形で設定される。

【0083】そのため機器間でオブジェクトが重複したIDを用いてもメッセージを正しく伝えることが可能になる。ボタン状態データはボタンが押されているか否かの状態を保持する。

【0084】図23は利用者がデジタルVTR203のアイコン表示229にカーソル230を合わせてダブルクリックした際の動作を示すフローチャートおよびコントロールパネルを操作した際の動作のフローチャートを示した図である。

【0085】図24は利用者でデジタルVTR203のアイコン表示229をダブルクリックした際のマルチメディアコントロール1の表示画面を示す図である。図24において231はデジタルVTR203のデフォルトのコントロールパネル表示画面、272は再生ボタンである。

【0086】図29はパネルラスのデジタルVTRコントロールパネルオブジェクトの構造とオブジェクト間の依存関係を示した図である。

【0087】図29において140はクラサスメソッドテーブルへのポインター格納部でありハネルクラスクラサスメソッドテーブル1402を示す。ハネルクラスクラサスメソッドテーブルはハネルオブジェクトを初期化するための面手段1404、バネルをダブルクリックされたときの動作を示すクリック反応手段1405から構成される。

1406はメッセージ通信手段、1407は処理待ち手段、1410は内部データ部であり、1411はオブジェクトID、1412はパネル状態データ、1413は描画パラメータである。内部データ部1410はデジタルVTR代理オブジェクトに紐づくデジタルファイル210の記述に従い、初期化されるが、デジタルVTR制御ユニット216にはデジタルファイル210のデジタルVTRコントロールロール情報1414、デジタルVTR記述部211はオブジェクト型情報1422、デジタルVTRアイコン画像1423、デジタルVTRコマンド情報1427を示す第2オブジェクトコントロール情報1424から成り立つ。オブジェクトコントロール情報1424はクラサ名1415（ハネルクラス）、オブジェクトID1416（ID=1）、所属オブジェクトIDから構成される。第1オブジェクト描画情報1418は描画位置・大きさ情報1419、形状・色情報1420、オブジェクト画像1421から構成され、第2オブジェクト画像1422は描画位置・大きさ情報1423、形状・色情報1424、オブジェクト画像1425から構成される。

【0088】図23と図29にしたがってデジタルVTR R R 203のコントロールパネル表示動作と再生動作の指示方法を説明する。図16で説明した動作においてシステムディレクトリオブジェクト205がデジタルVTR R 204でリアルタイムに生成された時点でデジタルVTR R 204はアイコン表示229をアイコン画像1426に基づいて提示するが、利用者がデジタルVTR R 204のアイコン229をカーソル230で指示し(643)、デジタルVTR R 204のコントロールパネルオブジェクト220のコントロールパネルオブジェクト221はコントロールパネルオブジェクト221を構成するすべてのオブジェクトにたいして描画を指示して図29に示すすべてのオブジェクトが描画手段817によってメッセージを送出する。該メッセージにしたがって図2









【0131】次に図2ではデジタルVTRの入力代理オブジェクトに対して、整合ファイルタイプを同一合わせ。本実施例のデジタルVTRは図45(b)に示す致のごく、整合ファイルタイプが複数あり、各ファイルタイプは優先順位とともに知られる。本実施例では図2のように、データ属性AudioMovieのファイルタイプAM4がVTR機器の最優先ファイルタイプとなっている。

【0132】次に、S3では図45(b)に示す致の優先順位に従って、図45(a)に示す致より一致するファイルタイプを検索する。図45(b)の致より、AM4が優先順位1であるので、図45(a)の致よりAM4フォーマットをサーチする。図45(a)の致にはAM4フォーマットがないので、優先順位2のAM5を図45(a)の致よりサーチする。

【0133】これも失敗した場合は、以下同様にして、一致するファイルタイプを優先順位に従ってサーチしてゆき、優先順位4のMovie2で始めて一致するファイルタイプが見つかることになり、これをデジタルカメラからデジタルVTRへのリンクにおける最速整合ファイルタイプとする。

【0134】S4では、S3でのサーチで一致するファイルタイプが見つかったので、S5へと処理を進める。

【0135】S5では、S3で得られた最速整合ファイルタイプであるMovie2のデータ属性Movie デジタルVTRの最優先ファイルタイプのデータ属性AudioMovieと一致していないのでS8へと処理を進める。

【0136】S8ではWindowServerに図34のような警告の表示を要求するメッセージを送知し、利用者からの入力待ちルーチンS12へと入る。

【0137】本実施例では利用者図34のパネル上でYESをクリックしたものと、S10よりS8へと処理を進める。

【0138】S2ではシステムデム内部データ部内の機器間リンク情報管理データ(図34の344)にデジタルカメラからデジタルVTRへとデータ属性Movie ファイルフォーマットMovie2の有効なリンクが張られたことを保存する。

【0139】最後にS7にて、デジタルカメラ出力代理オブジェクトとデジタルVTR入力代理オブジェクトに対してメッセージングし、デジタルカメラからデジタルVTRへとデータ属性Movie2のリンクが張られたことを通知し、データ入力管理手段43一連の処理を終了する。

【0140】S7にて、データ入出力管理オブジェクトよりメッセージを受けたデジタルカメラ出力代理オブジェクト(図41の410)はメソッド部のリンク情報更新手段(図41の413)を起動する。起動されたリンク情報更新手段(図41の413)は内部データ部のリンク情報(図41の419)に、Movie2フォーマットでデジタルVTR機器へとリンクが張られたことを格納するとともに対応データ入力オブジェクトIDであるデジタルカメラのデータ出力

することにより、デジタルVTRは310はデジタルカメラ301より自身に送られているMovie2フォーマットデータの記録を開始し、機器間のデータ転送が行われる。この時、マルチメディアコントロール内部と各機器間で送受されるメッセージングについて、以下に説明する。

【0141】まず、利用者がPlayボタン(図30の305)がクリックすると、WindowServerはクリップされた座標(コントロールパネル内での座標)をコントロールパネルオブジェクト(図32の333)に通知する。これを受けたコントロールパネルオブジェクトは自身の内部データ部を参照し、この座標にPlayボタンが表示されていることを知り、Playボタンがクリックされたことを解釈し、デジタルカメラ機器コントロールオブジェクト341に対して、利用者のマウスクリックを受けたデジタルカメラ機器の機械部分で、利用者からのデータ送出命令(PlayMessage)があったことを通知する。デジタルカメラコントロールオブジェクト(図43の429)内の処理機構手段431はメソッド部より、このメッセージに対応する再生実行手段433を起動する。再生実行手段433はデジタルカメラ機器のハードウェアを制御して、デジタルカメラ機器の機械部分をさらに再生可能な状態とする一方、デジタルカメラデータ出力オブジェクト(図31の403)に対して再生しているデータの出力要求メッセージを送信する。このメッセージを受けたデジタルカメラデータ出力オブジェクト内の処理機構手段(図42の421)はメソッド部よりデータ読み出し手段423及びデータ送信手段424を起動する。起動されたデータ読み出し手段は、記録媒体より情報を読み出す。

【0148】また、起動されたデータ送信手段は、まずリンク情報428を参照する。このとき、リンク情報428内の内容は図44の430のように、データ送信先オブジェクトID=120、ファイルタイプ=Movie2であるので、データ送信手段424は読み出されたデータをMovie2ファイルタイプにしたのち、ID=120に対してデータ送信を行う。実施例では、Movie2データは動画データであり、図38に示すように、データの先頭部分に、ファイルタイプや機械の画素数等の動画再生に必要なデータをヘッダ部分として持ち、本体部は、時間情報をしたタイムスタンプによって細分化されていて、このタイムスタンプ単位で編集作業を行うことができるものである。

【0149】ID=120のデジタルVTR機器データ入力オブジェクト(図27の208)はID=110のオブジェクトより、Movie2データが送信されてきたことを検知すると、自身のリンク情報(図27の1030)を参照する。図44の431のように、ID=110のデータ出力オブジェクトとのMovie2ファイルタイプでのリンクがあることを確認すると、送信されてくるデータのヘッダ部分(図36の364)をメモリ(図37の370)に覚えておく。

【0150】利用者がある時点で、デジタルVTRコントロールパネルのRecボタン(図30の318)をクリックすると、WindowServerはクリップされた座標(コントロール

ールパネル内での座標)をコントロールパネルオブジェクト(図31の328)に通知する。これを受けたコントロールパネルオブジェクトは自身の内部データを参照し、この座標にRecButtonが表示されていることを知り、RecButtonがクリックされたことを解釈し、デジタルVTR機器コントロールオブジェクト341に対して、利用者マウスクリックを受けたデジタルVTRコントロールオブジェクトからのデータ記録命令(BeMessage)があったことを通知する。これを受けたデジタルVTRコントロールオブジェクト内の処理機構手段(図15の1011)はメソッド部より録画実行手段(図15の1020)を起動する。起動された録画実行手段(図15の1020)はデジタルVTR機器のハードウェアを制御して、デジタルVTR機器の機械部分をさらに再生可能な状態としたのち、デジタルVTRデータ入力オブジェクト(図27の208)に対して、入力されているデータの記録要求を示すメッセージを送信する。データの記録要求を受けたデジタルVTRデータ入力オブジェクト(図27の208)はデジタルVTR機器のハードウェアを制御し、RAM(図37の370)に保存されているヘッダ情報(図36の364)を読み込み、ヘッダ情報の後に、次に送信されてくるタイムスタンプ以降のデータを繋ぎ合わせて、超伝導記録媒体374への記録を開始する。このようにして記録されたMovie2動画データは動画再生に必要なヘッダ情報をもつて再生が可能となる。

【0151】《第二の実施例》以下、本発明の第2の実施例について説明する。本実施例におけるマルチメディアコントロールは図1の実施例の特徴を有すると共に、アプリケーションソフト(ユーティリティソフト)動作環境を有する。

【0152】以下本実施例におけるマルチメディアコントロールについて説明する。(本発明においては、アプリケーションソフトとユーティリティソフトは本質的に差異はなく、以下ユーティリティソフトも含めてアプリケーションソフトと呼ぶとする)。

【0153】図39に本発明の第2の実施例におけるマルチメディアコントロールの内部のオブジェクトの構成を示す。本発明の第2の実施例におけるマルチメディアコントロールは、マルチメディアコントロールのアプリケーションソフト(ユーティリティソフト)インストール手段及びアプリケーションソフトの動作環境を有する。(本発明においては、アプリケーションソフトとユーティリティソフトは本質的に差異はなく、以下ユーティリティソフトも含めてアプリケーションソフトと呼ぶとする)。

【0154】本実施例のマルチメディアコントロールにおいては、アプリケーションソフトはフロッピーディスクや通信手段等からマルチメディアコントロール内のアプリケーションクラスタ格納部(図39の388)にインストールされる。アプリケーションソフトの内容はオブジェクト指向に

【0141】まず、利用者がPlayボタン(図30の305)がクリックすると、WindowServerはクリップされた座標(コントロールパネル内での座標)をコントロールパネルオブジェクト(図32の333)に通知する。これを受けたコントロールパネルオブジェクトは自身の内部データ部を参照し、この座標にPlayボタンが表示されていることを知り、Playボタンがクリックされたことを解釈し、デジタルカメラ機器コントロールオブジェクト341に対して、利用者のマウスクリックを受けたデジタルカメラ機器の機械部分で、利用者からのデータ送出命令(PlayMessage)があったことを通知する。デジタルカメラコントロールオブジェクト(図43の429)内の処理機構手段431はメソッド部より、このメッセージに対応する再生実行手段433を起動する。再生実行手段433はデジタルカメラ機器のハードウェアを制御して、デジタルカメラ機器の機械部分をさらに再生可能な状態とする一方、デジタルカメラデータ出力オブジェクト(図31の403)に対して再生しているデータの出力要求メッセージを送信する。このメッセージを受けたデジタルカメラデータ出力オブジェクト内の処理機構手段(図42の421)はメソッド部よりデータ読み出し手段423及びデータ送信手段424を起動する。起動されたデータ読み出し手段は、記録媒体より情報を読み出す。

【0148】また、起動されたデータ送信手段は、まずリンク情報428を参照する。このとき、リンク情報428内の内容は図44の430のように、データ送信先オブジェクトID=120、ファイルタイプ=Movie2であるので、データ送信手段424は読み出されたデータをMovie2ファイルタイプにしたのち、ID=120に対してデータ送信を行う。実施例では、Movie2データは動画データであり、図38に示すように、データの先頭部分に、ファイルタイプや機械の画素数等の動画再生に必要なデータをヘッダ部分として持ち、本体部は、時間情報をしたタイムスタンプによって細分化されていて、このタイムスタンプ単位で編集作業を行うことができるものである。

【0149】ID=120のデジタルVTR機器データ入力オブジェクト(図27の208)はID=110のオブジェクトより、Movie2データが送信されてきたことを検知すると、自身のリンク情報(図27の1030)を参照する。図44の431のように、ID=110のデータ出力オブジェクトとのMovie2ファイルタイプでのリンクがあることを確認すると、送信されてくるデータのヘッダ部分(図36の364)をメモリ(図37の370)に覚えておく。

【0150】利用者がある時点で、デジタルVTRコントロールパネルのRecボタン(図30の318)をクリックすると、WindowServerはクリップされた座標(コントロール

基づいたクラスの記述ファイルであり、システムディレクトリ内のアプリケーションオブジェクト生成手段(図39の386)はこのクラスを用いることにより、アプリケーションソフトの実行可能なオブジェクトをアプリケーションオブジェクト制作エリア(図39の389)に生成することによりアプリケーションソフトは実行可能な状態となる。アプリケーションオブジェクトがマルチメディアコントロールデータに描画する可視化されたウィンドウオブジェクト上で利用者からのマウスのポインティング手段による操作は、WindowsServerにより、ウィンドウ上のどの座標でどのような操作が行われなかを通知するメッセージとして、アプリケーションオブジェクトに伝えられる。アプリケーションオブジェクトは自分の内部データに、ウィンドウ上のどの座標にどのようなグラフィックスが存在するのといった情報を保有しているもので、WindowsServerからの情報を聞いて、アプリケーションウィンドウ上のアイコンのクリックやアイコン間のドラッグ動作等を認識することができ、[0155]以下、マルチメディアコントロールのアプリケーションソフトの具体例の一つとして、コネクションストラクタアプリケーションについて説明する。

[0156] 図38はコネクションストラクタのユーザインタフェースを示すものである。図38において、370はコネクションストラクタウィンドウ、379は利用者の入力手段であるマウスのカーソル、371は376及び383、それぞれ、CD-Player、VTR、ディスプレイ、CA-TV、デコダ、DAT、デッキ、音声信号を増幅し図示されていないスピーカより音声を生じさせるアンプ、光磁気ディスクの再生記録をおこなうMO-Player、マルチメディアコントロール、及びUD-Playerのアイコンである。各機器間のデータ送受信関係(コネクション)は矢印によって示めされ、利用者はこれらの矢印により、容易に機器間のデータ送受信関係を知ることができる。

[0157] 本実施例ではUD-Player(図38の383)にはコネクションが張られていないが、このような状態でUD-Playerを再生状態とすると映像及び音声データはデフォルトのリンク先であるところのマルチメディアコントロール内の映像表示及び音声出力を司るオブジェクトへと送られる。

[0158] 以上のごとく表示されたコネクションストラクタウィンドウは、コネクションの表示のみではなく、ウィンドウ上でコネクションの編集作業を行うことができる。利用者はマウスにより各アイコン間をドラッグすることにより、コネクションをはることができ、図38では利用者からマウスをドラッグすることによりCD Player(371)よりアンプ(376)へとコネクションを張る。[0159] 利用者がマルチメディアコントロールデータ

る為、曲等を用いてもいいようにかまわない)。また、それぞれのリンクにおいて有効データの属性は線の種類(または色)によって区別されている。

[0164] 本実施例では、図38のように、Visualデータは映像、Audioデータは点線、アプリケーションプログラムは一点線、そしてテキストデータは2点線にそれぞれ表示されている。

[0165] たとえば、図38では、VTR機器のアイコン372とDisplay機器のアイコン373が実際の矢印379で結ばれているが、これはVTR機器を再生すればVTR器は自動的にディスプレイ機器へと送られることを意味している。また、利用者がCA-TVを見たい時には、利用者はCA-TVのコントロールパネルを開きチャンネルを設定するので、CA-TVの映像データは自動的にディスプレイ機器373のID宛てに送信され、またCA-TVの音声データは自動的にアンプ機器376のID宛てに送信される。

[0166] また、本実施例では、MO-Playerからのデータはプログラムデータであり、プログラムデータはマルチメディアコントロールへと転送されると、マルチメディアコントロールはデータのヘッダ部分に記述されているデータ属性にプログラムと記述されているのを知り、これをアプリケーションプログラムと認識し、自動的にこのプログラムが起動される。この機能はマルチメディアコントロールに限らず他の機器でも実現できる。例えばディスプレイ機器内にOperatingSystemを設け、ディスプレイ機器がプログラムを受け取った際には、ディスプレイ機器がプログラムを実行するようにする。

[0167] 本実施例ではこのプログラムは内部に映像データが埋め込まれていて、このプログラムを実行することにより、ディスプレイ上に映像が映し出される。この方式の利点は映像データがプログラムに埋め込まれている為、機器間でデータのフォーマットを認識せずにすむことである。

[0168] このようなウィンドウ上で、利用者はマウス等の入力手段によりウィンドウ上に表示されている各機器のアイコン間リンクを張ることにより、各機器間のデータ送受信関係を編集することができ、アイコン間にコネクションが張られると、コネクション編集手段394はシステムディレクトリオブジェクトに対して、LinkMessageを送信する。LinkMessageを受け取ったシステムディレクトリオブジェクトは実施例1と同様に図38のフローに準って処理を行う。システムディレクトリ内データ入出力管理手段によりリンクが有効であると判断されたときには、リンクに関連するオブジェクトの内部データは実施例1と同様に更新される。

[0169] また、コネクションストラクタオブジェクトは図38の397で、システムディレクトリオブジェクトより張られたリンクのデータ属性を通知され、コネクションストラクタオブジェクトはこのデータ

属性に合わせてリンクの形態を選択し、表示するので、リンクの形態に適合するオブジェクトの内部データとコネクションストラクタウィンドウ上での表示とは常に一致することとなる。

[0170]

[発明の効果] 以上の様に、本発明によるシステム機器制御方式によれば、複数のマルチメディア機器が接続されたシステム全体の制御を行なう際に、今までにあらがれなかったコントロール側にその制御を行なうためのデバイスドライバやアプリケーションソフトウェアなどをインストールして準備する必要がなくなり、マルチメディア機器をLAN上に接続するだけで自動的にコントロールパネル及び機器状態がコントロールの画面上に表示され、電源ON/OFF、本体の制御・入出力の切り替えを等インタフェースをもつて、マルチメディア機器間のデータの送受信関係を構築することができる。また、データ送受信の際のファイルフォーマットの問題はマルチメディアコントロールが自動的に行うので、利用者はファイルフォーマットの問題といった面倒な作業を行わずにすむ。

[図面の簡単な説明]

[図1] マルチメディアコントロールとマルチメディア機器の物理的接続形態を示す図である。

[図2] マルチメディアコントロールとマルチメディア機器の物理的接続形態を示す図である。

[図3] オブジェクト化されたマルチメディア機器の内部構造を示す図である。

[図4] オブジェクト化されたマルチメディアコントロールの内部構造を示す図である。

[図5] マルチメディアコントロールのシステム階層図である。

[図6] マルチメディア機器のシステム階層図である。

[図7] マルチメディア機器をマルチメディアコントロールに接続する前の状態を示す図である。

[図8] LANにマルチメディア機器が接続されたときの状態を示す図である。

[図9] 一般的なクラスライブラリーの状態を示す図である。

[図10] オブジェクトの状態を示す図である。

[図11] システムディレクトリオブジェクトの構造を示す図である。

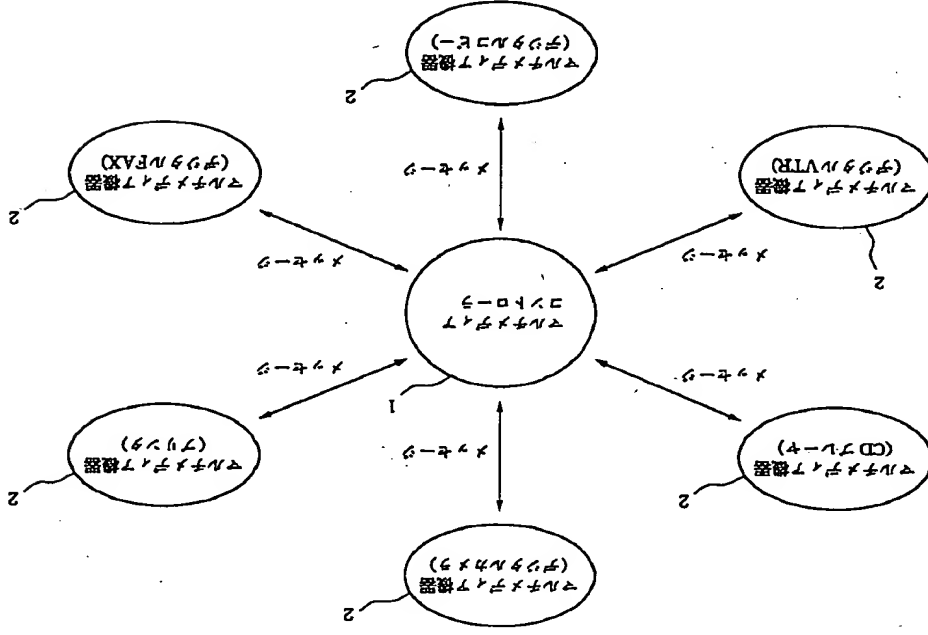
[図12] 代理オブジェクト記述ファイルのコントロールパネル記述部の構成を示す図である。

[図13] 代理オブジェクト記述ファイルのデータ入出力オブジェクト記述部の構成を示す図である。

[図14] オブジェクト化されたデジタルVTRをマルチメディアコントロールに接続する前の状態を示す図である。

[図15] VTRコントロールオブジェクトの構造を示

【図1】



マルチメディアコントロールとマルチメディア機器の接続形態図

図である。

【図31】本発明におけるマルチメディアコントロール、VTR機器及びデジタルカメラ機器の内部オブジェクトの構成と、各機器の接続状況を示すものである。

【図32】データ入力管理手段の処理手順を示すフローチャートである。

【図33】デジタルカメラからデジタルVTRへリンクを張った際の、データ出力管理手段の処理手順のフローチャートである。

【図34】警告パネルのデザインを示す図である。

【図35】デジタルカメラ出力代理オブジェクトのメニュー部及び内部データ、デジタルVTR入力代理オブジェクトのメニュー部及び内部データをそれぞれ示す図である。

【図36】動画データのデータ形式を示す図である。

【図37】デジタルVTR機器の内部構成を示す図である。

【図38】コネクションコンストラクターウィンドウのユーザーインターフェースを説明するための図である。

【図39】本発明の第2の実施例におけるマルチメディアコントロール内部のオブジェクトの構成を致すブロック図である。

【図40】コネクションコンストラクターオブジェクトのコネクションコンストラクターウィンドウ表示手段の処理フローを示す図である。

【図41】デジタルカメラ出力代理オブジェクトの内部の構造を示す図である。

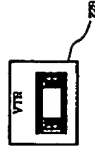
【図42】デジタルカメラデータ出力オブジェクトの内部の構造を示す図である。

【図43】デジタルVTR Cameraコントロールオブジェクトの内部の構造を示す図である。

【図44】デジタルカメラよりデジタルVTRへとリンクを張った後の各オブジェクト内部データ部を示す図である。

【図45】デジタルカメラ統合ファイルリスト及びデジタルVTR統合ファイルリストをそれぞれ説明するための図である。

【図18】



す図である。

【図16】デジタルVTRをLANに接続した際の動作のフローチャートである。

【図17】マルチメディアコントロールの画面を示す図である。

【図18】LANにマルチメディア機器としてオブジェクト化されたデジタルVTRが接続されたときの状態を示す図である。

【図19】デジタルVTRのアイコンを示す図である。

【図20】コントロールパネル画面を示す図である。

【図21】オブジェクトの所属するクラスとデジタルVTRコントロールパネルオブジェクトの構成要素の対応を説明する図である。

【図22】再生ボタンオブジェクトの生成に関する説明図である。

【図23】デジタルVTRのアイコン表示にカーソルを合わせて制御を選択した際の動作のフローチャートである。

【図24】操作者がデジタルVTRのアイコン表示から制御を選択した際のマルチメディアコントロールの表示画面を示す図である。

【図25】デジタルVTRデータ入力代理オブジェクトの構造とオブジェクト記述情報の関係を示した図である。

【図26】デジタルVTRデータ出力代理オブジェクトの構造とオブジェクト記述情報の関係を示した図である。

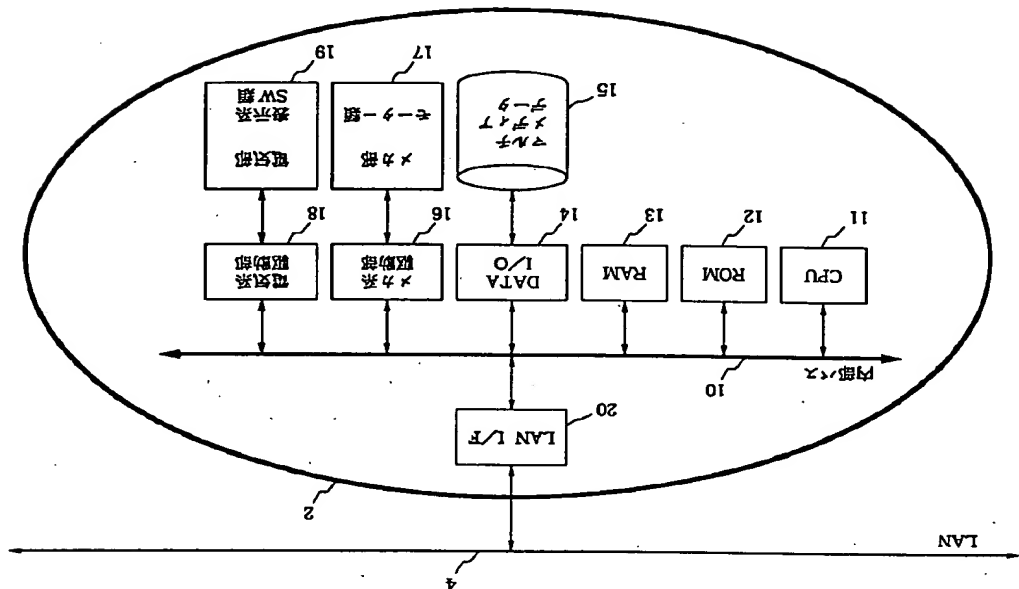
【図27】デジタルVTRデータ入力オブジェクトの構造を示す図である。

【図28】デジタルVTRデータ出力オブジェクトの構造を示す図である。

【図29】パネルクラスのデジタルVTRコントロールパネルオブジェクトの構造とオブジェクト記述情報の関係を示した図である。

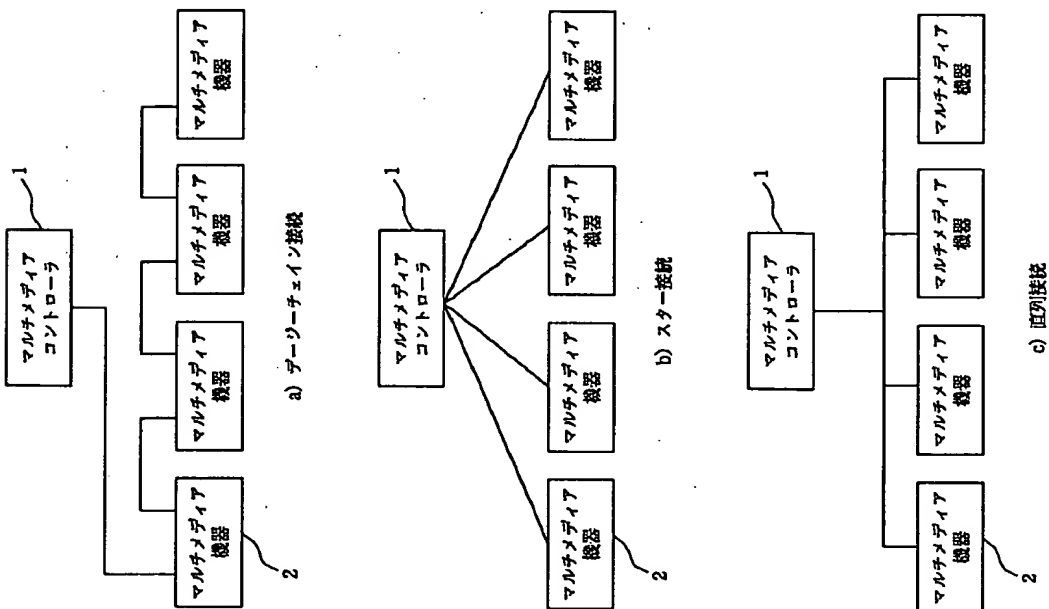
【図30】マルチメディアコントロールメニューディスプレイ上に表示されるコントロールパネル及びその動作を示す

【図3】



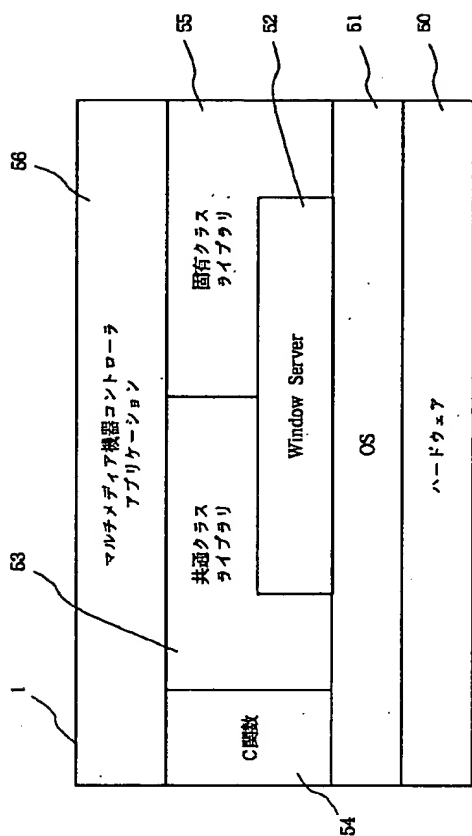
マルチメディア化したマルチメディア機器の内部構造

【図2】

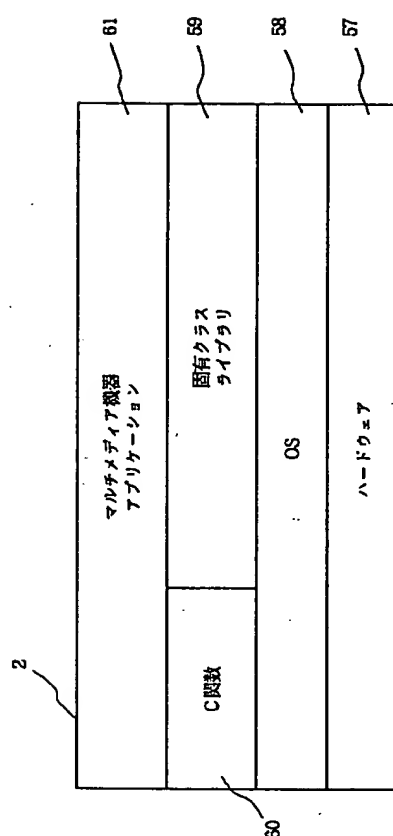


マルチメディアコントローラとマルチメディア機器の物理的接続形態

【例5】

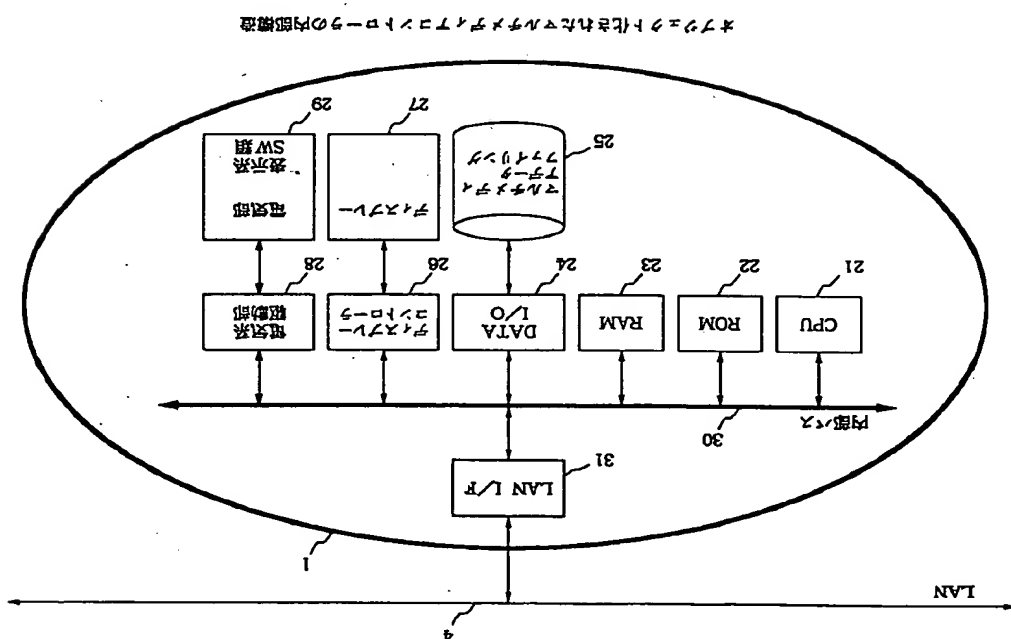


マルチメディアコントロールのシステム階層図



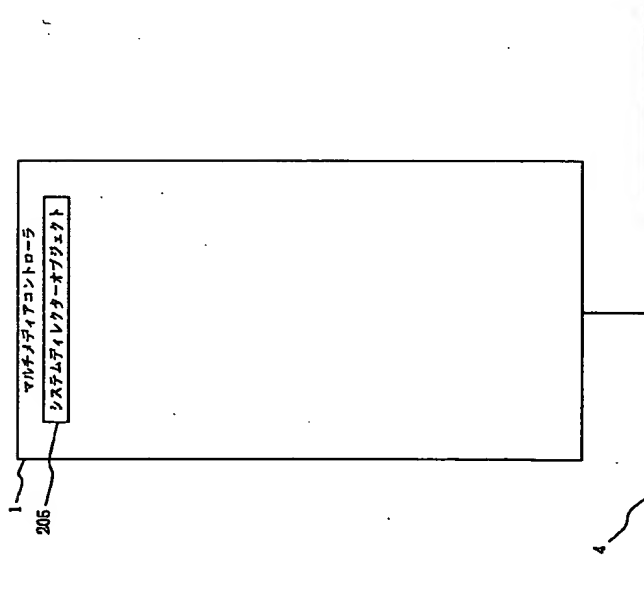
マルチメディア機器のシステム階層図

【図4】

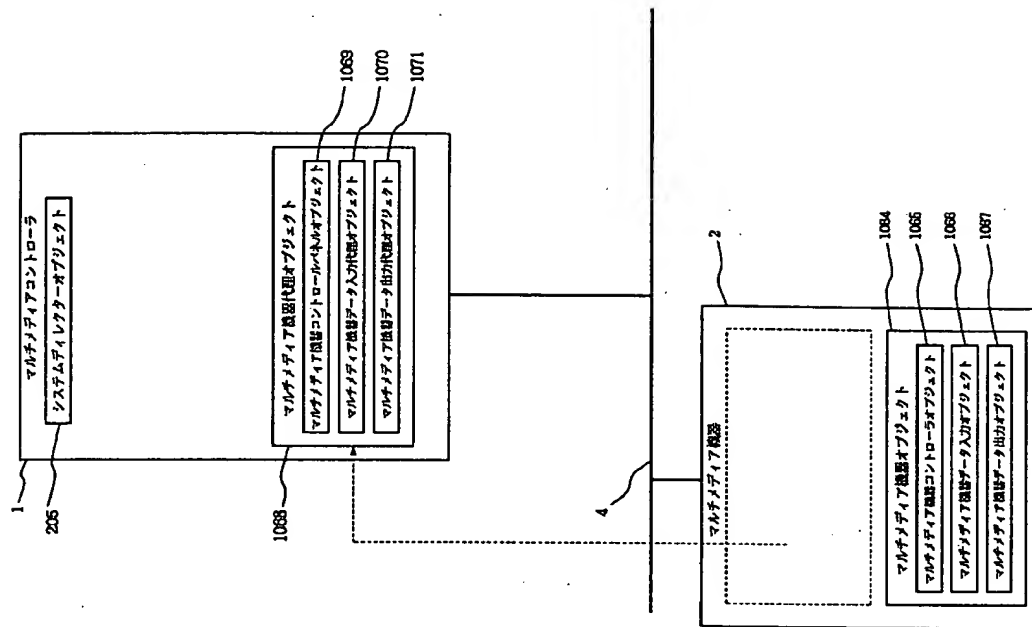




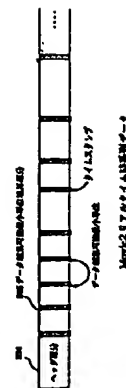
【図7】



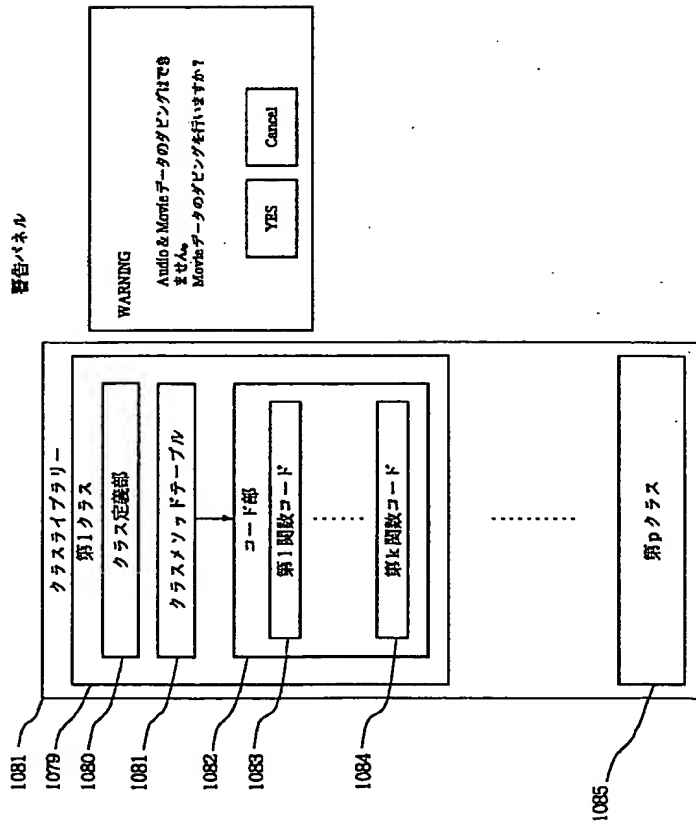
【図8】



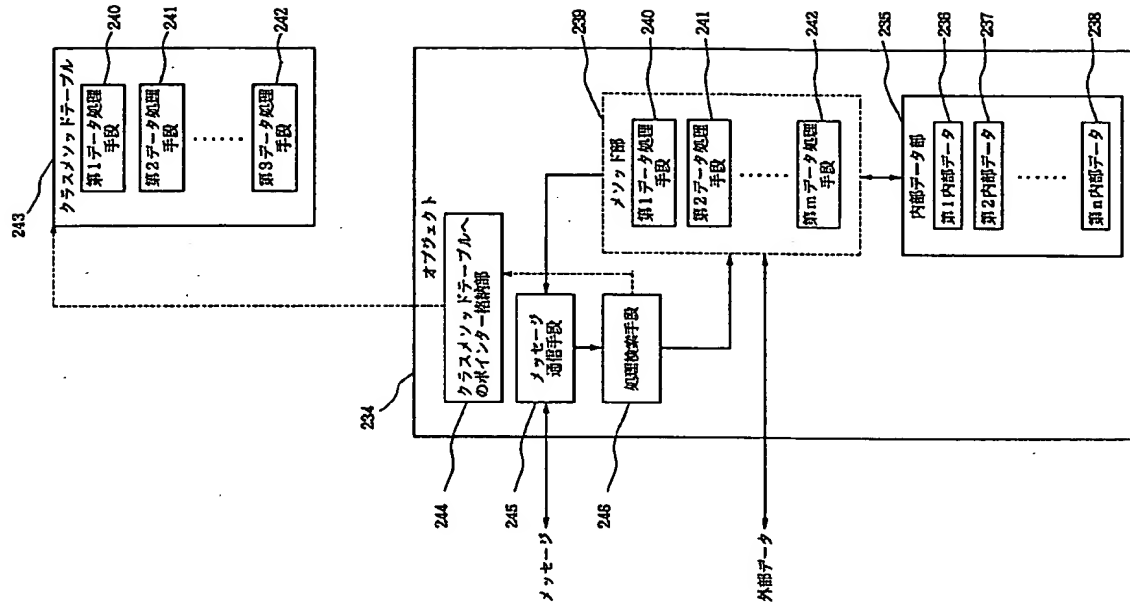
【図36】



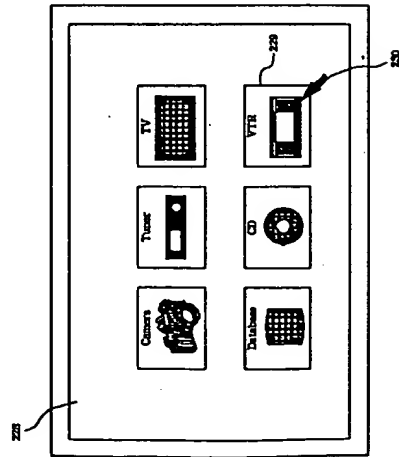
【図 9】



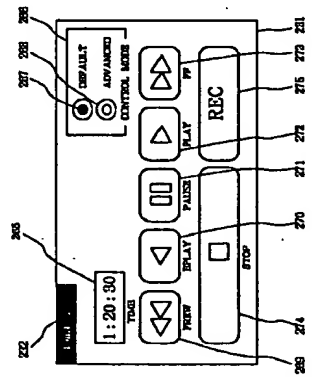
【図 10】



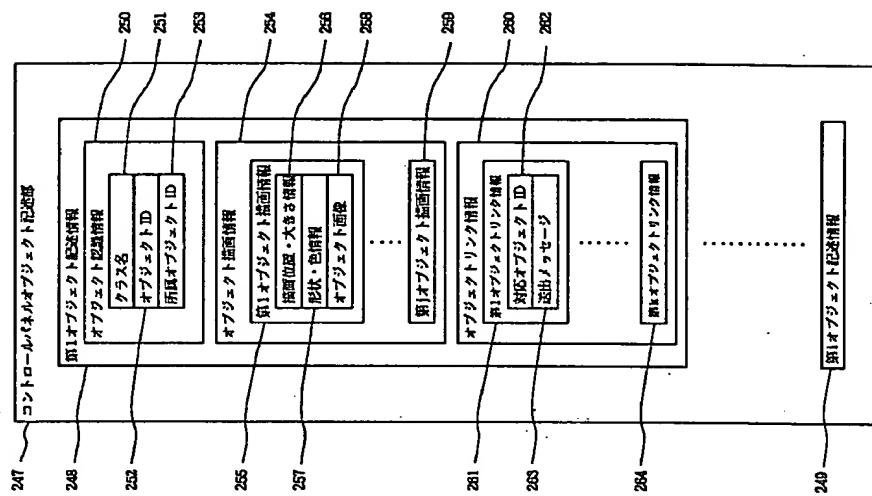
【図 17】



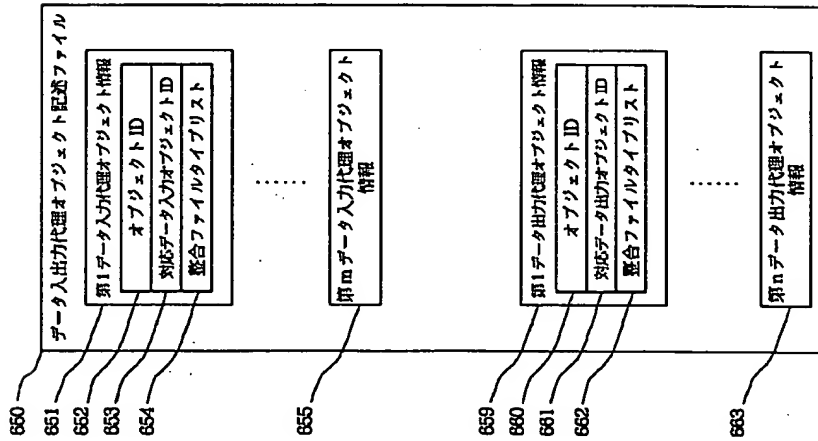
【図 20】



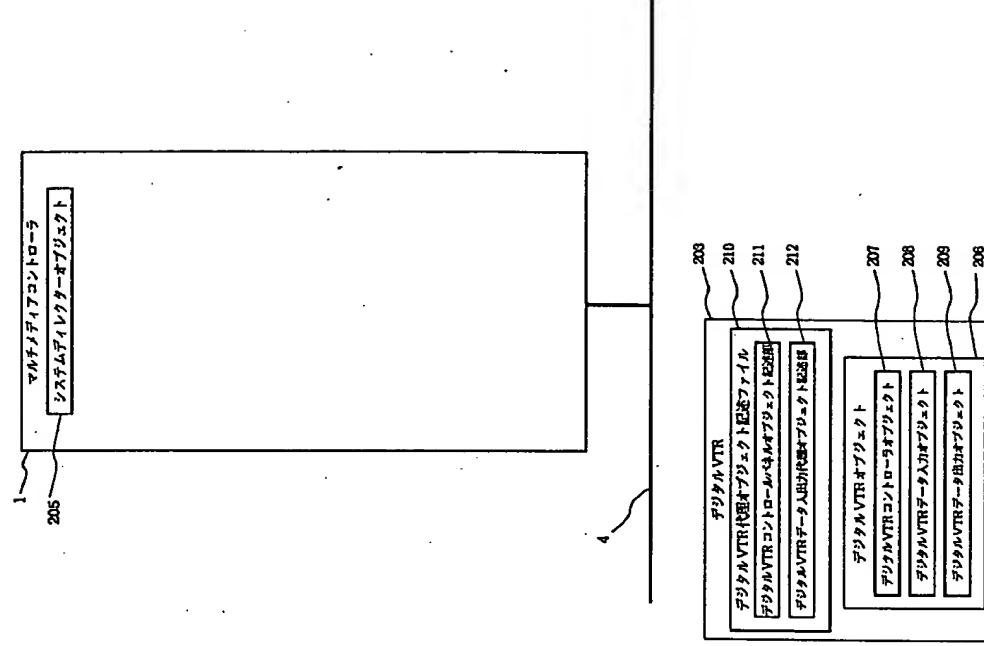
【圖12】



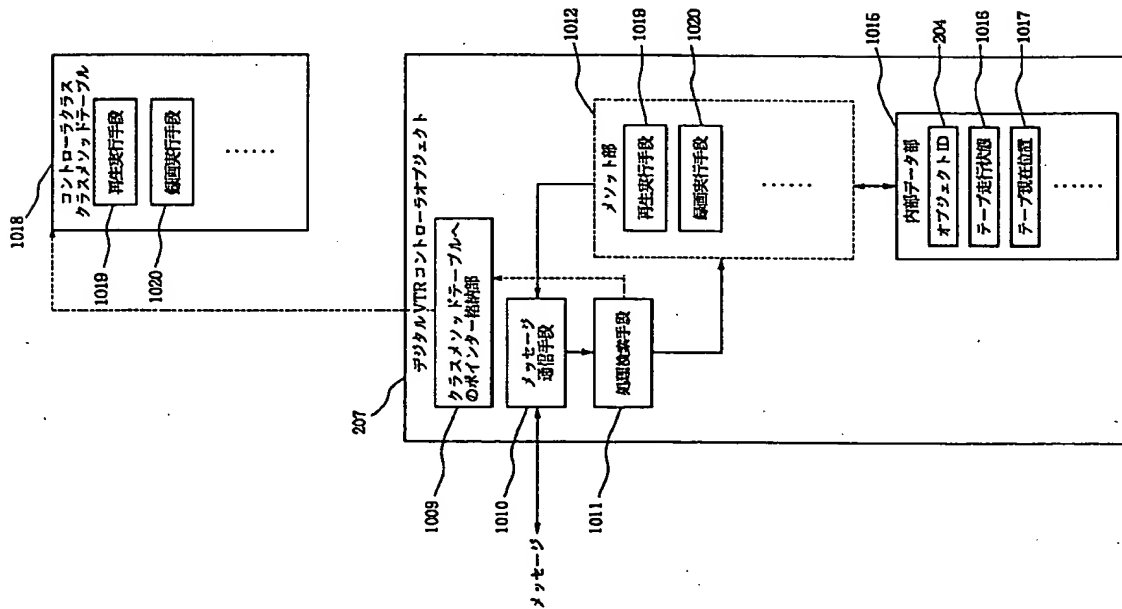
【図13】



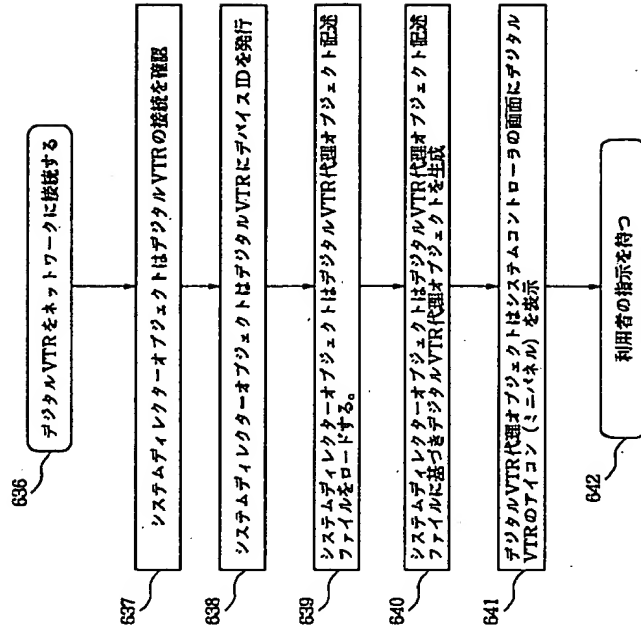
【図14】



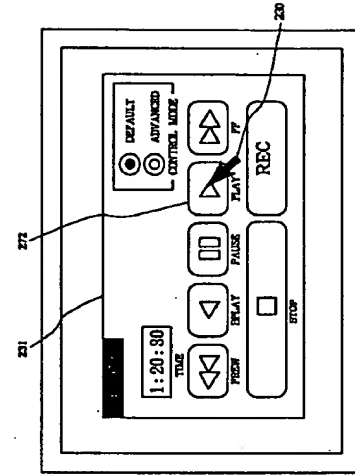
【図15】



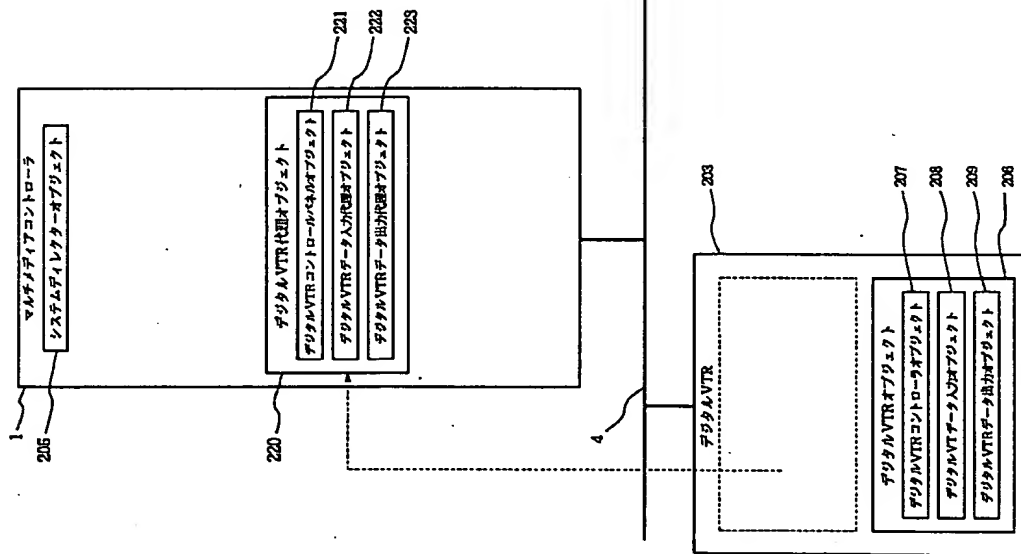
【図16】



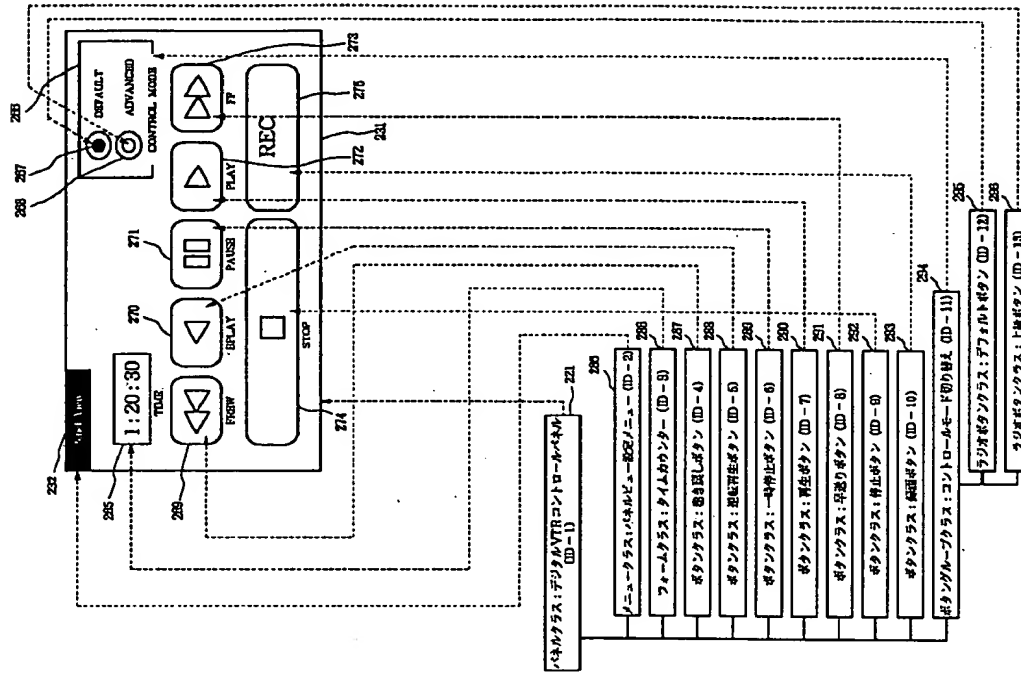
【図24】



【図 18】

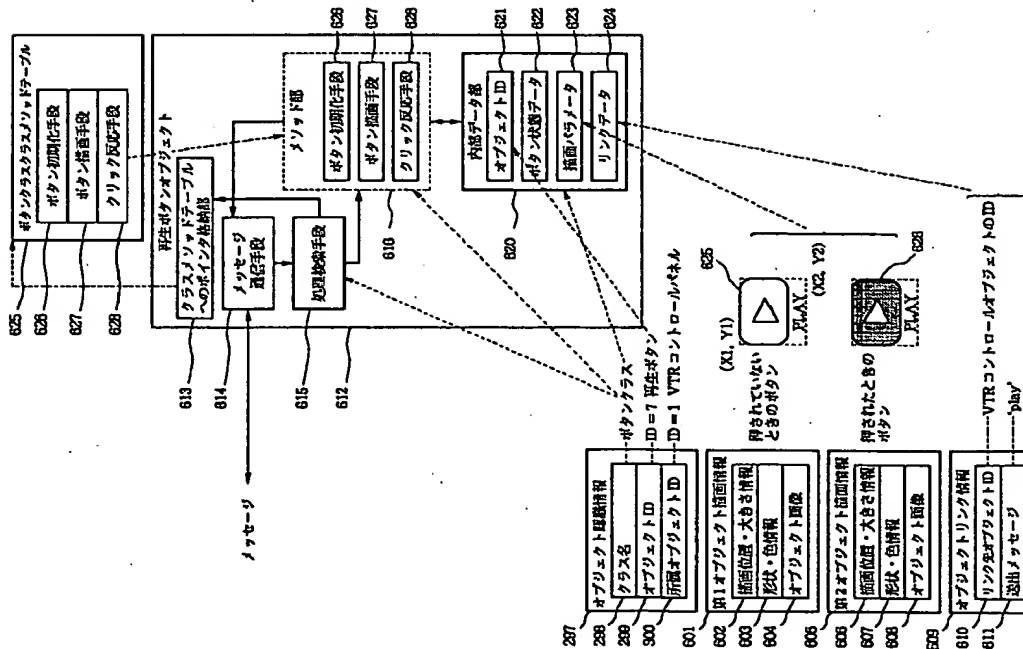


【図 21】

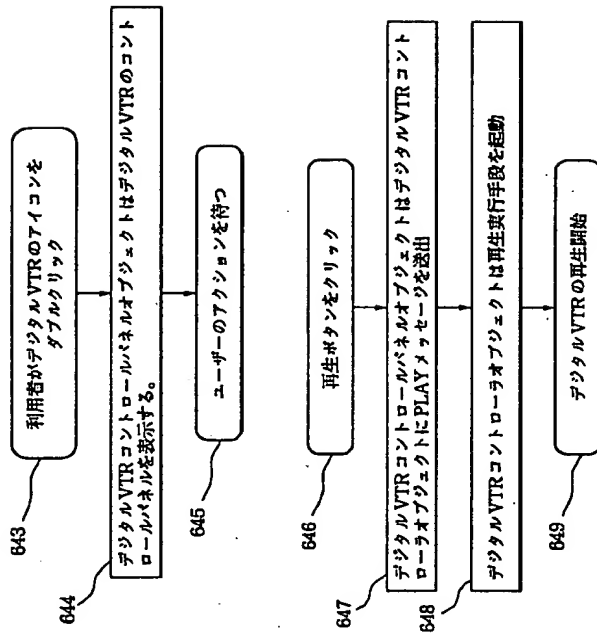




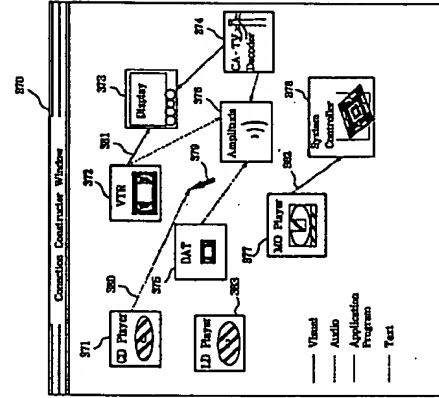
【图22】



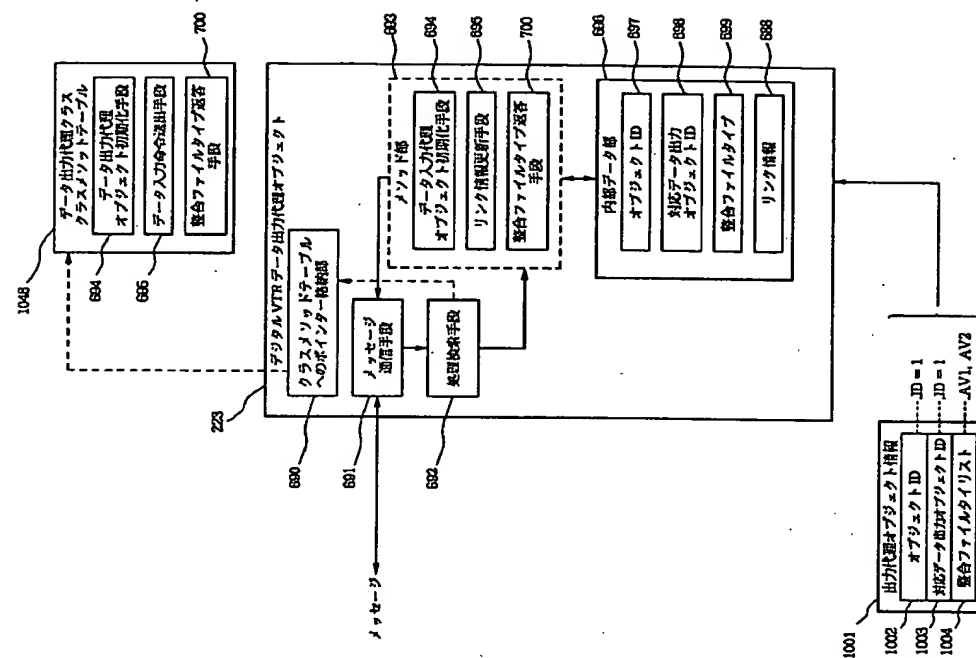
**【图23】**



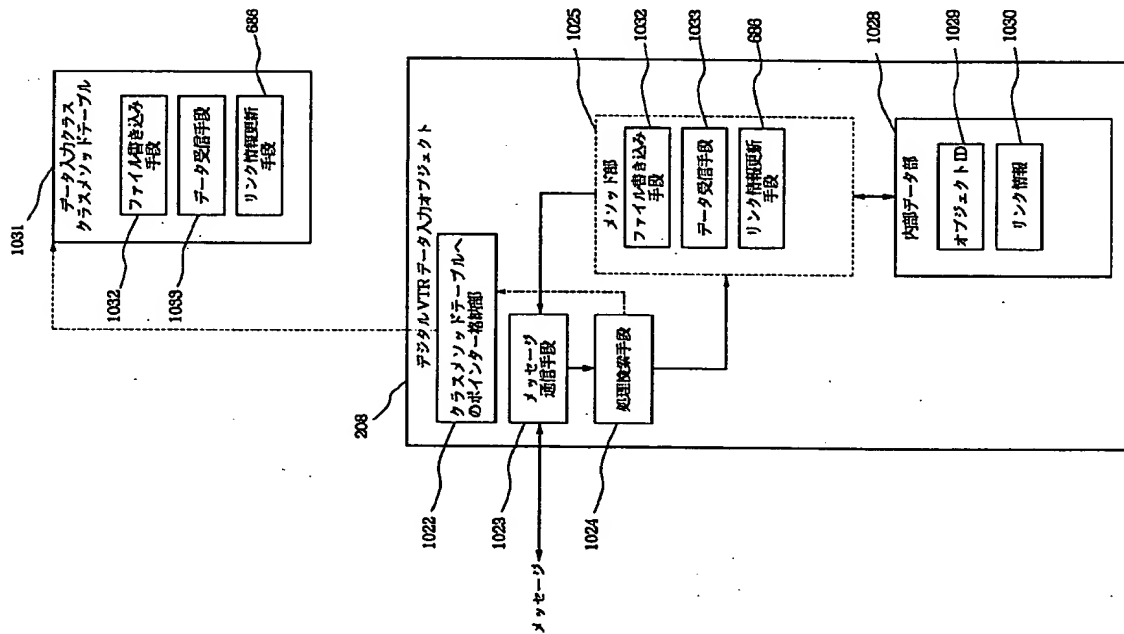
【图38】



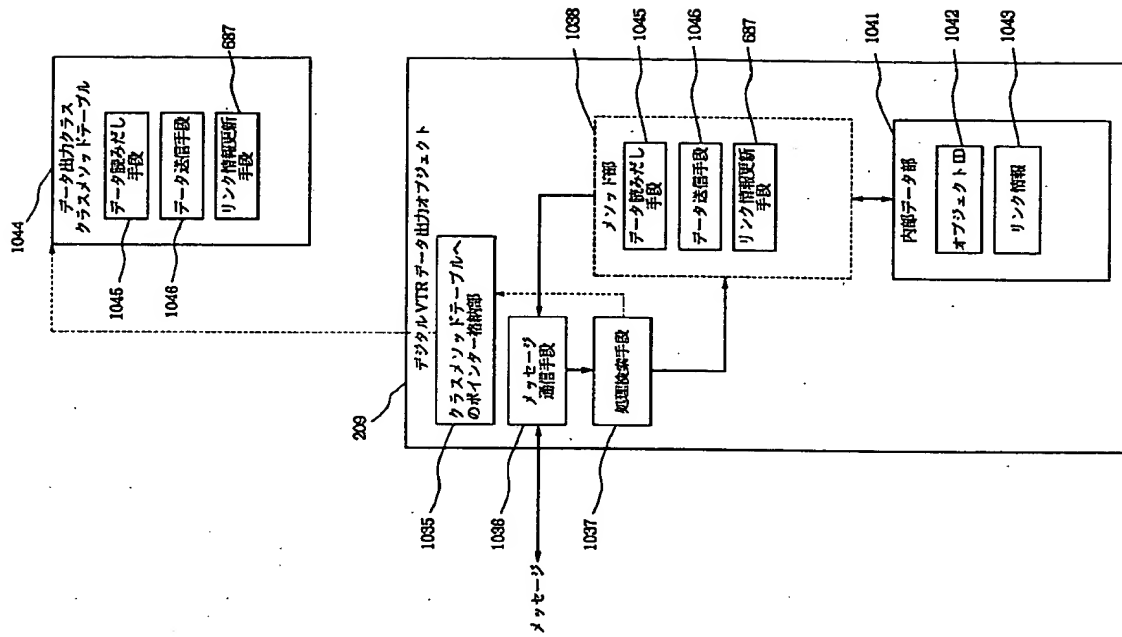
【图26】



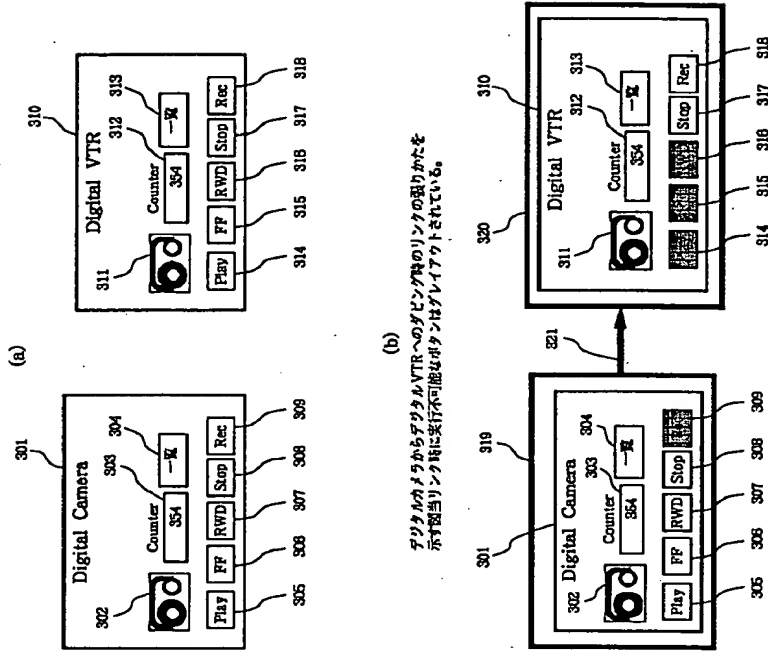
【図 27】



【図 28】



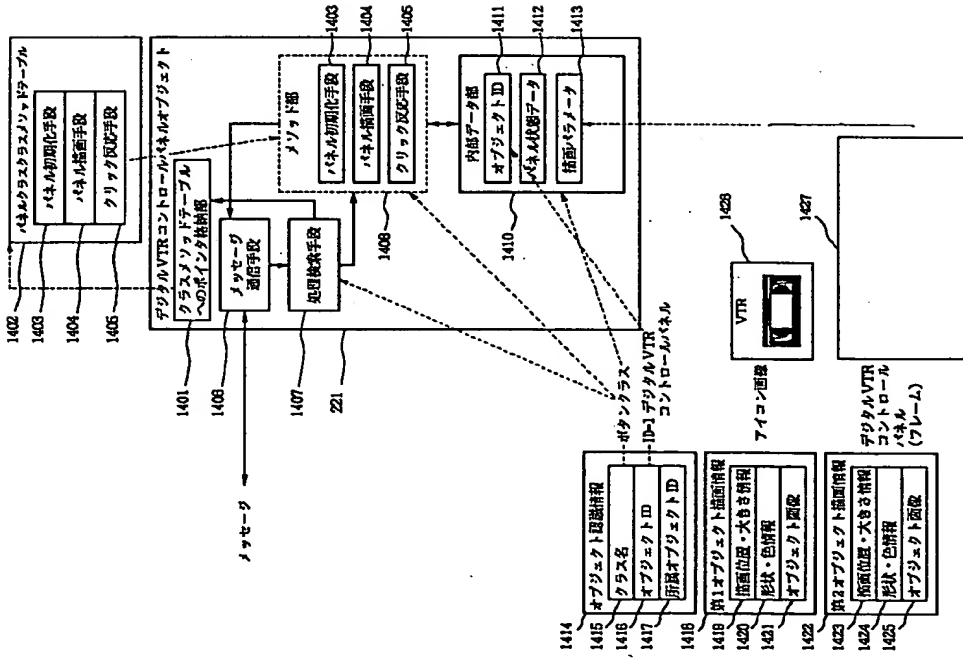
【図 30】



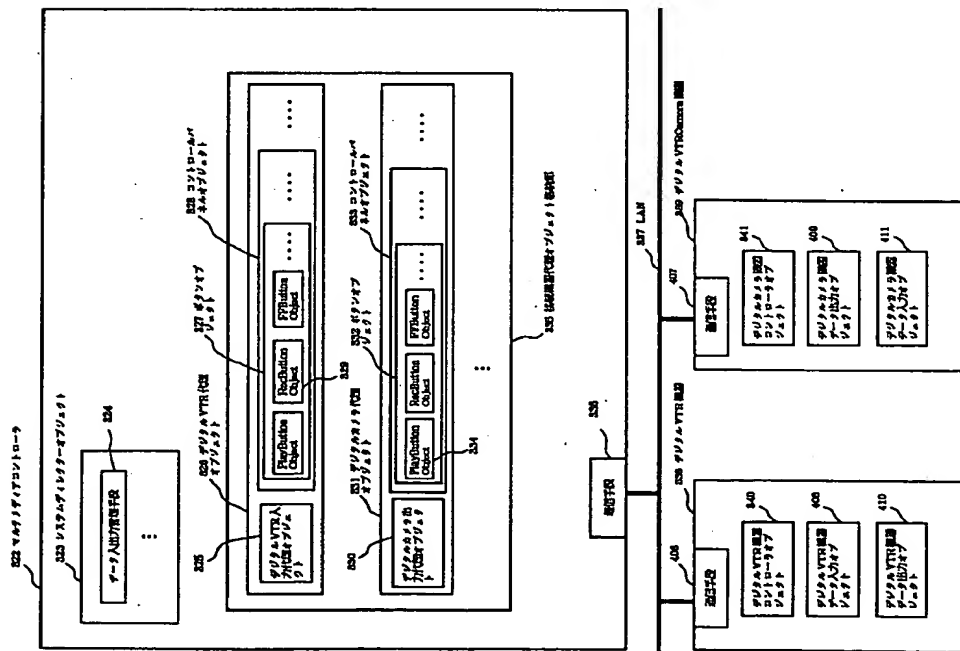
デジタルカメラからデジタルVTRへのデジタライズ時のリンクの張りかたを示す図(当リンク時に実行可能なボタンはフレイクされている)

マルチメディアコントロールディスプレイ上の装置間の接続の指定方法を示す図

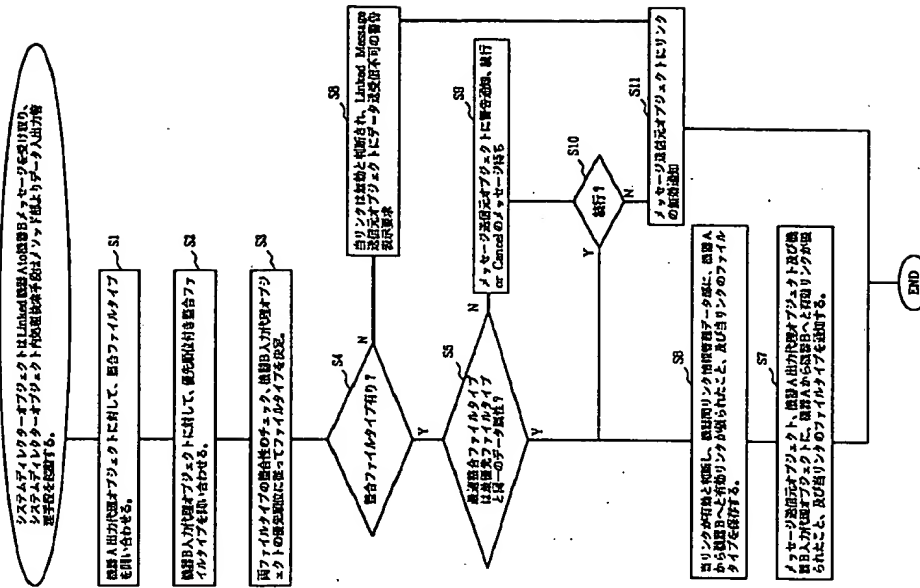
【図 29】



【图31】

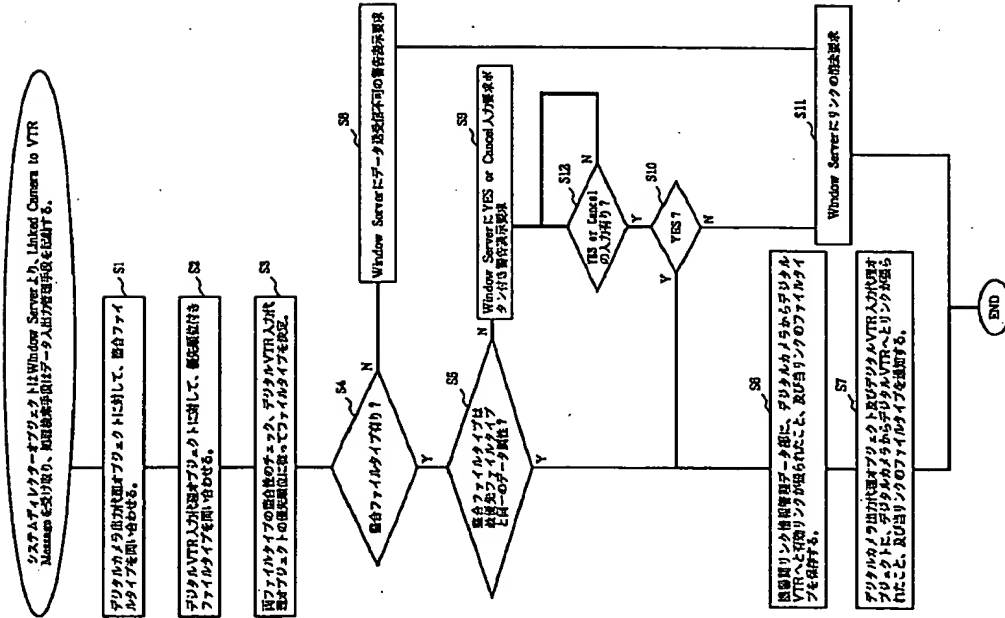


**[圖32]**



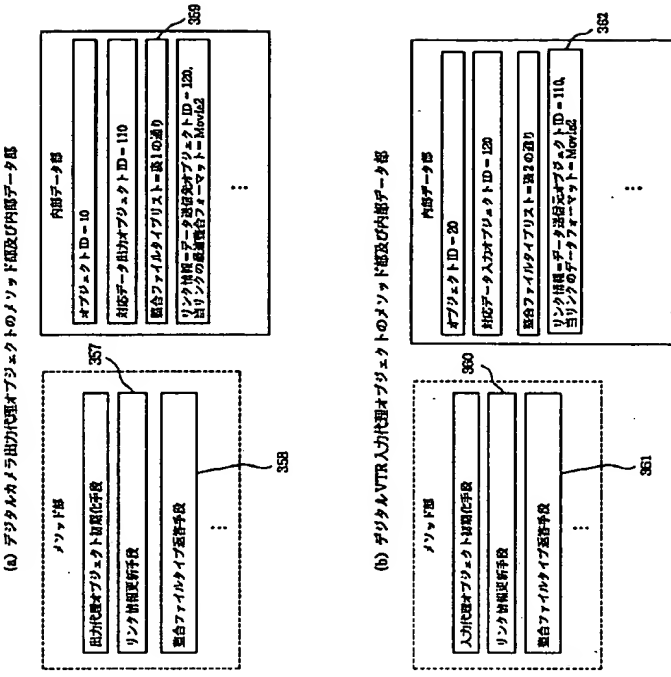
データ入出力管理手段の処理手順を示すフローチャート

【図 33】



デジタルカメラからデジタルVTRへリンクを張った際、データ出力管理手段の処理手順フローチャート

【図 35】



【図 45】

(a) デジタルカメラ統合ファイルタイプリスト

データ属性	ファイルタイプ
Audio	Audio1, Audio2, Audio3
Movie	Movie1, Movie2
Audio & Movie	AM1, AM2, AM3

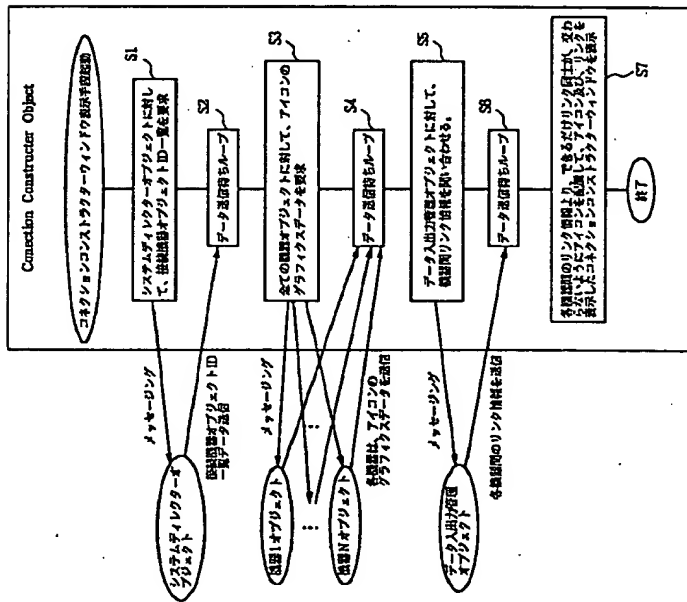
(b) デジタルVTR統合ファイルタイプリスト (優先順位付き)

データ属性	優先順位付きファイルタイプ
Audio	③ Audio3, ⑦ Audio4, ⑧ Audio5
Movie	⑤ Movie1, ④ Movie2
Audio & Movie	① AM4, ② AM5, ③ AM6

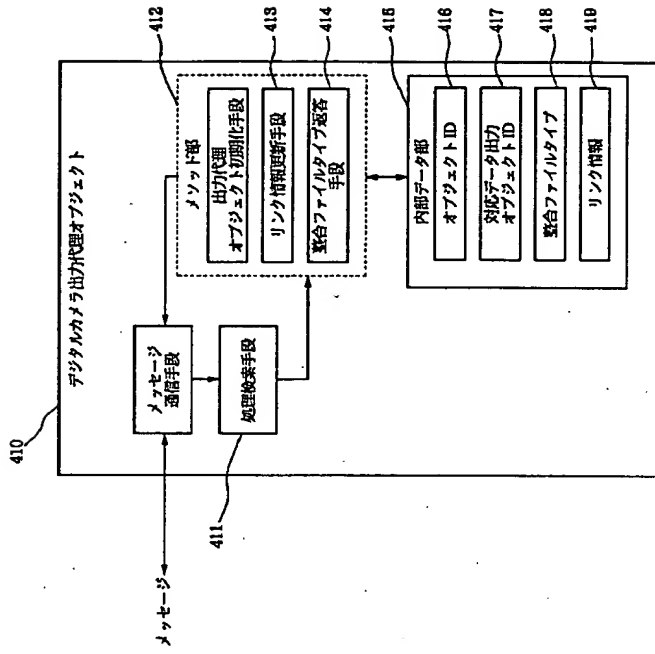




【図40】

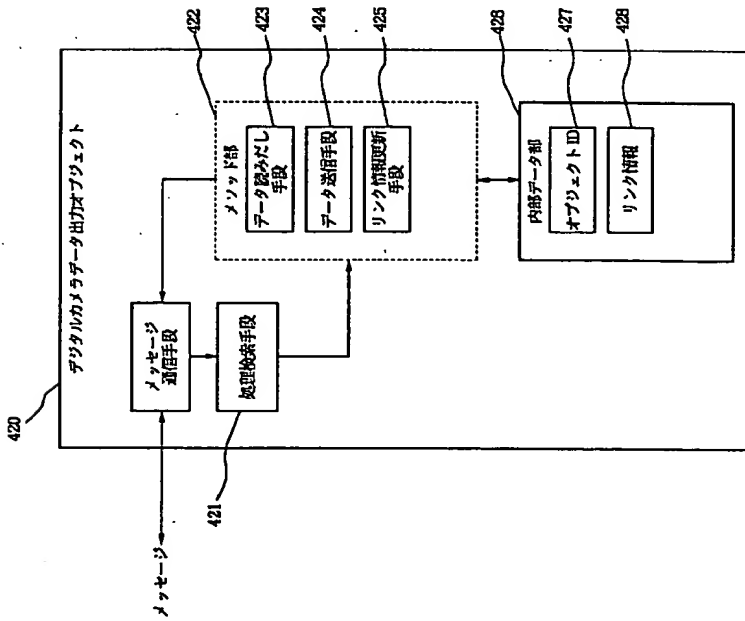


【圖41】

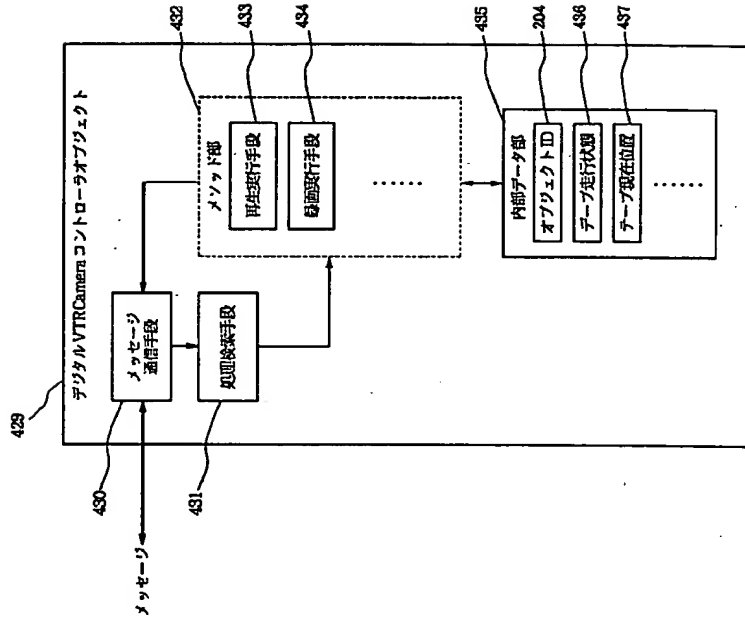


コネクションコンストラクタ-オブジェクトのコネクションコンストラクタウィンドウ表示手段の処理フロー

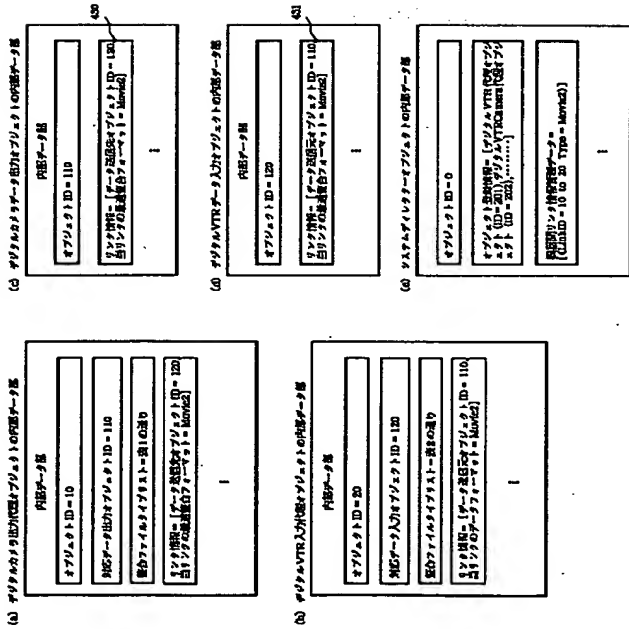
【図 4 2】



【図 4 3】



【図44】



プログラムの実行順序は、プログラムの実行順序に従って決定される。

フロントページの続き

	識別記号	片内整理番号	F I	技術表示箇所
(51)Int.Cl. <sup>4</sup>				
H 0 4 L 12/28				
H 0 4 N 5/00	A			
5/445	Z			
5/765				
5/907	B 7734-5C			
// G 0 6 F 9/44	5 3 0 M 9193-5B			
12/00	5 4 7 D 8944-5B			
(72)発明者 河村 秀明				(72)発明者 相澤 隆志
東京都大田区下丸子3丁目30番2号キヤノン株式会社内				東京都大田区下丸子3丁目30番2号キヤノン株式会社内
(72)発明者 河村 秀明				(72)発明者 羽島 健司
東京都大田区下丸子3丁目30番2号キヤノン株式会社内				東京都大田区下丸子3丁目30番2号キヤノン株式会社内

TRANSLATION

From column 4, line 4 to column 34, line 21

[0012]

[Embodiment]

The embodiment of the present invention is explained in detail below with reference to the drawings.

[0013]

First, in the present invention, a system control method is employed in which the multimedia devices are individually recognized as objects and the controller manages the objects in a comprehensive fashion.

[0014]

Each object has a function to send its own function/control means to the controller in order to be managed by the controller. Consequently, it is no longer necessary to install a control program in the controller in advance, such that control capability may be obtained simply by connecting the device to the controller.

[0015]

The controller has a means to have the control means sent from the connected object be displayed and operated by the person who actually issues the control instructions. Consequently, the controller can manage the multimedia devices in a concentrated fashion, and can achieve sufficient flexibility and expandability to respond to a new multimedia device without going through a new preparation session.

[0016]

Incidentally, since the object-orientation concept itself used in the present invention is explained in detail in the reference literature, such as in Ishizuka, Object-Oriented Programming, Aschy Publishing, 1988'; Sakai, Introduction to Object Orientation, Aum, 1990'; and B. J. Cox, Object-Oriented

Programming, Toppan, 1988, for example, a basic technical explanation will be omitted in the explanation of the embodiment of the present invention provided below.

[0017]

Object-oriented technology has been drawing attention in recent years from the viewpoint of creating a more efficient program development environment, but it may also be widely utilized for operating systems and multimedia databases. There are three characteristic concepts in object-oriented technology:

(1) Encapsulation

(2) Succession

(3) Messaging

The present invention seeks to perform development and expansion based on these concepts so that they may be applied in the control of multimedia devices.

[0018]

The present invention uses a method in which audiovisual devices and a controller are connected across a network, and the audiovisual devices are individually recognized as objects while the controller manages the objects in a comprehensive fashion. Once an audiovisual device is connected to the network, it is given an object ID that is used as the transfer destination when an object-oriented message and data are sent to it. By performing transfer of data based on this object ID, the destination of the data is logically determined. Therefore, it is not necessary to change the physical wiring when creating and modifying the data transfer relationships among the audiovisual devices, and instead, only the internal object data related to the creation and modification of the data transfer relationship should be changed.

[0019]

In addition, each object has a function to send its own

function/control means to the controller in order to be managed by the controller. Consequently, it is no longer necessary to install a control program in the controller in advance, such that control capability can be obtained simply by connecting the device to the controller. The controller has a means to have the control means sent from the connected object be displayed and operated by the person who actually issues the control instructions. As a result, all audiovisual devices connected to the network may be operated by means of one interface. Further, a means to manage the connections among the devices is located inside the controller, and by having this managing means display the connection status of each device using the display means, the user can easily recognize the connection status even if many connection relationships have been created.

[0020]

In addition, using the operating means, the user can perform editing of the connections. By messaging the contents of the editing to the managing means, the managing means understands the intention of the user, asks the objects that possess the information regarding the data I/O for each device for the information regarding data I/O in order to determine the data compatibility between devices, and displays the result of this determination on the display means. Consequently, the user can easily learn whether or not data can be transferred between devices.

[0021]

Further, because the display means displays the objects symbolizing each audiovisual device and the user creates a connection relationship by connecting these displayed objects by means of the operating means, an erroneous connection cannot be created by the user.

[0022]

Fig. 1 shows a logical form of connection between a

multimedia controller and multimedia devices, in which the object-orientation concept of the present invention is incorporated. With the multimedia controller 1 in the center, each multimedia device 2 has an established communication path to the controller so that direct communications may be made regarding various types of information on a one-on-one basis. Control is performed by sending messages back and forth via the communication path. The multimedia devices specifically comprise any device that handles multimedia data, including AV equipment such as CD players, digital VTRs, digital cameras and digital TVs, and OA equipment such as digital facsimiles, digital copying machines and printers.

[0023]

For the controller, a dedicated machine is assumed here. However, it may also be achieved by installing a dedicated OS and an application software program in a personal computer, or in a general purpose computer in a word processor work station.

[0024]

Fig. 2 shows the various possible physical connection configurations to establish two-way communication paths between the multimedia controller and the multimedia devices as (a) through (c).

[0025]

Fig. 2(a) shows the daisy chain connection method adopted for a SCSI bus (ANSI X3. 131-1986), Fig. 2(b) shows the star connection method adopted by Ethernet (IEEE 802.3) 10BaseT, and Fig. 2(c) shows the serial connection method adopted by Ethernet, 10Base2/5.

[0026]

A combination connection configuration method such as GPIB (IEEE 488) in which configurations (a) through (c) are combined may also be used. Ethernet also uses a method combining configurations (b) and (c). For the communication method,



methods using optical or ISDN cables are available, and various combinations or options other than those shown in Fig. 2 may also be used.

[0027]

The present invention does not specifically address how the two-way communication path should be established or which method should be used. However, while physical limitations due to the difference in communication methods (such as transfer speed, number of connected devices, length of connection and connector configuration) do not pose obstacles to two-way message communication on account of the difference in protocol hierarchy, in order to reliably perform mutual connection of the peripheral devices, the devices must share an interface having at least one physical (mechanical or electrical) characteristic in common.

[0028]

Moreover, in order to realize high-speed data communication such as for moving images, a method using optical communication such as FDDI (Fiber Distributed Data Interface) or B-ISDN, both of which are faster than Ethernet, may be employed. In this application, however, for the sake of explanation, it is assumed that the devices use Ethernet 10Base2(/T), which is widely available at an inexpensive price, as the common communication connector.

[0029]

Fig. 3 shows a general internal block diagram of the hardware of a multimedia device.

[0030]

The multiple multimedia devices are connected to the controller via the LAN 4. Since the LAN here comprises an Ethernet network, the device has an interface 20 that processes the Ethernet protocol (TCP/IP). This can be obtained through the use of a dedicated LSI chip, etc. In this interface, the

received message itself is extracted, or conversely, a message is sent to the controller. As an example of the message, the general form is expressed as follows in Objective-C.

[0031]

[Target object method name: arguments]

While the expression may differ in other languages, it is identical in principle, and the following designations are made.

[0032]

(1) Designation of the target object

(2) Designation of the method (the process to be executed)

(3) Designation of the arguments (parameters), if any

The handling of this message is explained in the software sequence shown in Fig. 3.

[0033]

Inside the multimedia device are a CPU 11 that performs all software processes and hardware control via the internal bus 10, a ROM 12 in which programs, initial values and unique information are stored, a RAM 13 in which temporary data and internal parameters such as the device status are stored and which is used as a work area during the execution of a program, a data I/O 14 that accesses the multimedia data 15 stored in an internal or external storage medium, a mechanical drive unit 16 that controls the mechanisms 17 such as the motor, and an electrical drive unit 18 that controls the electrical members 19 of the display system, such as the switches SW and the LED. The multimedia data 15 is the area in which the digital data for images, sounds, letters, etc. is stored. It can take various forms, including an optical disk such as a CD-ROM or MD, a magnetic tape medium such as DCC or DAT, or a semiconductor memory card.

[0034]

Fig. 4 shows an internal block diagram of the hardware of

the multimedia controller. The controller is connected to the multimedia devices through the LAN 4 in the drawing. Since the LAN uses Ethernet, the controller has an interface 31 that processes the Ethernet communication protocol (TCP/IP). This can be obtained through the use of a dedicated LSI chip, etc. In this interface, the received message itself is extracted, or conversely, a message is sent to the multimedia devices.

[0035]

Inside the multimedia controller are a CPU 21 that performs all software processes and hardware control via the internal bus 30, a ROM 22 in which programs, initial values and unique information are stored, and a RAM 23 in which temporary data and internal parameters such as the device status are stored and which is used as a work area during the execution of a program. The multimedia filing device 25 performs storage, searching, reproduction, editing, etc. of the multimedia data regardless of whether the data is stored internally or externally. The data I/O 24 performs access control regarding the multimedia filing device 25. An electrical drive unit 28 that controls the electrical members 29 of the display system, such as the switches SW and the LED, a display 27 that comprises a machine interface, a display controller 26 that performs the display control for the display 27 and a pointing device comprising a mouse, etc., not shown in the drawing are also included.

[0036]

Fig. 6 shows the system hierarchy of the multimedia device in terms of software. The internal block diagram shown in Fig. 3 constitutes the hardware 57. The OS 58 performs the basic control to control the hardware. While no particular OS is required, it is preferable for it to have real-time capability and a multitasking function to execute multiple programs simultaneously. Each multimedia device has a unique class

library 59 on this OS in order to perform object-rendering of the multimedia control of the multimedia device.

[0037]

Further, while not shown in the drawing, the multimedia device has a control panel and a control library so that it may be controlled by the controller. By sending this library when the device is connected to the controller, control operations unique to the multimedia device may be performed on the side of the controller. The multimedia device also has a C function 60 that performs a timer function and calculation.

[0038]

In the top layer resides the application software 61 that governs the control of the multimedia device, the communication with the multimedia controller and the user interface. Based on exchanges of messages with the controller, this application enables various types of control and program operations with regard to the multimedia device, which is recognized as a single object, and the internal parameters may be read and changed as instance variables.

[0039]

Fig. 5 shows the system hierarchy of the multimedia controller in terms of software. The internal block diagram shown in Fig. 4 constitutes the hardware 50. The OS 51 executes the basic control to control the hardware. While no particular OS is required here again, it is preferable for it to have real-time capability and a multitasking function.

[0040]

On top of the OS resides a window server 52 that runs all GUI (graphical users interface) functions such as display of the control screen regarding the connected multiple multimedia devices, display of the overall system connection status and switching of data I/O. In the common class library 53 is stored a group of basic common parts (a group of objects) regarding

the user interface or control operations, such as buttons, a sliding volume indicator and a text display area, which are prepared by the controller initially.

[0041]

Conversely, in the unique class library 55 is stored a group of parts (a group of objects) related to the panel display and to control operations unique to the connected multimedia devices. This unique library is sent from a multimedia device and increases in size each time a multimedia device is connected to the system, as described above. The specific procedure followed here is explained below. There is also a C function 54 that performs the timer function and calculation. At the top layer is the application software 56 that is responsible for the control of all connected multimedia devices, as well as for communication with the multimedia devices and the user interface.

[0042]

The specific control sequence and the exchange of messages between the controller and a multimedia device will now be explained.

[0043]

Fig. 7 is a drawing showing the situation before the multimedia device is connected to the multimedia controller. In Fig. 7, 4 is the LAN to perform digital data communication and 1 is the multimedia controller that controls the operation of the entire system. 2 represents a generalized construction of the multimedia devices connected to the LAN 4. 205 is a system director object, which is a software object (hereinafter abbreviated as 'object') that resides in the multimedia controller 1 at all times and performs management of the entire system.

[0044]

1064 is a multimedia device object, which is an object that

functions as a object-rendered multimedia device with regard to other objects on the LAN 4. The multimedia device object 1064 further comprises three objects 1065, 1066 and 1067.

[0045]

1065 is a multimedia device controller object that performs control of the hardware in order to execute most of the functions of the multimedia device 2, 1066 is a multimedia device data input object that is responsible for the input of digital data from other devices via the LAN 4, and 1067 is a multimedia device data output object that is responsible for the output of digital data to other devices via the LAN 4.

[0046]

1061 is a multimedia device substitute object description file that describes the specifications of the multimedia device substitute object that is generated inside the multimedia controller 1 when the multimedia device 2 is connected to the multimedia controller 1 via the LAN 4. The multimedia device substitute object description file 1061 comprises a multimedia device control panel object description 1062 that describes the specifications of the operation panel for the multimedia device 2 and a data I/O substitute object description 1063 that describes the specifications of the data I/O substitute object that performs substitution in the data I/O regarding the multimedia device 2. In particular, the multimedia device control panel object description executes the GUI description language function to describe the control panel for operating the multimedia device 2 using the GUI.

[0047]

Fig. 8 is a drawing to explain the situation in which the multimedia device 2 is connected to the LAN 4. In Fig. 8, 1068 is an object generated inside the multimedia controller 1 and is a multimedia device substitute object 1068 that functions inside the multimedia controller 1 as a substitute for the

multimedia device 2. The multimedia device substitute object 1068 comprises a multimedia device control panel object 1069 that functions as the control panel for the multimedia device 2, a multimedia device data input substitute object 1070 that functions as a substitute for the multimedia device data input object 1066 during data input, and a multimedia device data output substitute object 1071 that similarly functions as a substitute for the multimedia device data output object 1067. [0048]

Fig. 9 is a drawing showing the generalized construction of a class library. In Fig. 9, 1079 is a first class, which is one of the classes that define the characteristics and functions of objects having a similar nature, and which functions as a template for the generation of an object. A group of p number of classes, i.e., from the first class 1079 to the pth class 1085, is called a class library 1086, and all objects belong to a particular class. 1080 is a class definition unit that defines the data type and name of the internal variables possessed by the objects belonging to the class and the data type and name of the internal function that indicates the data processing means (generally termed 'class method'), 1081 is a class method table in which the pointers for each code of the class method are indicated in the form of a table in order to enable access to the class method, and 1082 is a code unit that stores a total of k number of class method function codes from the first function code 1083 through the kth function code 1084. [0049]

Fig. 10 is a drawing showing the generalized construction of an object. In Fig. 10, 234 is an object, and comprises a class method table pointer storage unit 244, a message communication means 245, a process search means 246, a method unit 239 and an internal data unit 235. The method unit 239 comprises a total of m number of data processing means from the first data

processing means 240 and the second data processing means 241 to the mth data processing means 242. 235 is an internal data unit and comprises a total of n number of internal data items from the first internal data item 236 and the second internal data item 237 to the nth internal data item 238. [0050]

Each internal data item comprising the internal data unit 235 is unique to each object and therefore it resides inside its own object, but because the data processing means of the method unit may be shared between objects if they share the same class, the data processing means, i.e., the first data processing means 240 through the mth data processing means 242, are managed by class via the class method table 243 and are shared by multiple objects belonging to the same class. The class method table 243 is referred to by each object by means of the pointer stored in the class method table pointer storage unit 244. [0051]

The message communication means 245 receives messages from other objects and sends them to the process search means 246. The process search means 246 analyzes the message and searches in the method unit 239 (actually, in the class method table 243) for a data processing means that corresponds to the message and causes it to execute processes. The data processing means executes prescribed processes regarding the data attached to the message, internal data residing in the internal data unit 235 and external data. Depending on the process, a message may be sent out to another object. In that case, the message is sent to the other object via the message communication means 245. [0052]

Fig. 11 is a drawing showing the construction of the system director object 205. In the drawing, 1072 is a class method table pointer storage unit and points to the system director class

class method table 1073. 1047 is a multimedia device substitute object generating means that generates a multimedia device substitute object 1068 based on the description of the multimedia device substitute object description file 1061. 343 is a data I/O managing means that manages the data I/O between objects, and 380 is an application object generating means that generates application objects for various purposes. 1074 is a message communication means, 342 is a process search means, and 1075 is a method unit. 1076 is an internal data unit, object ID, 344 is inter-device link information management data [regarding the links between devices] when a certain operation is performed using multiple multimedia devices, and 1078 is object registration information regarding the connected multimedia devices and the generated objects.

[0053]

When the multimedia device 2 is connected to the LAN 4 using the multimedia device substitute object generating means 1047, the system director object 205 reads the multimedia device substitute object description file 1061, selects the class to which the object that should be generated from the information described in the multimedia device substitute object description file 1061 belongs, and generates a multimedia device substitute object 1068 based on the class definition unit 1080 of the applicable class in the class library 1081.

[0054]

Fig. 12 is a drawing showing the construction of the control panel description of the substitute object description file. In Fig. 12, 247 is a control panel object description and comprises a total of  $i$  number of items of object description information from the first object description information item 248 to the  $i$ th object description 249. One object description information item comprises object recognition information item 250, object draw information item 254, and object link

information item 260.

[0055]

The object recognition information item 250 comprises a class name 251 that indicates the class to which the object belongs, an object ID 252 which is a unique ID for the  $i$ th object, and an upper object ID 253 that indicates the ID of the object to which the  $i$ th object belongs.

[0056]

The object draw information item 254 is the information to perform drawing of the objects such as buttons that form the control panel display screen 231, and comprises a total of  $j$  number of items of object draw information from the first object draw information item 255 to the  $j$ th object draw information item 259. One object draw information item comprises a draw position/size information item 256, a shape/color information item 257, and an object image item 258.

[0057]

The object link information item 261 is a description that provides information regarding the links with the objects to which the objects comprising the control panel object correspond, such as the controller object 207, and comprises a total of  $k$  number of items of object link information from the first object link information item 261 to the  $k$ th object link information item 264. One object link information item comprises a corresponding object ID 262 and a message 263 sent to the corresponding object.

[0058]

Fig. 13 shows the construction of the data I/O substitute object description of the substitute object description file. In Fig. 13, 650 is a data I/O substitute object description, 651 is the first input substitute object information item and 655 is the  $m$ th input substitute object information item. Each input substitute object information item comprises its own

object ID 652, a link destination corresponding data input object ID 653 that indicates the ID of the link destination data input object, and a compatible file type list 654, which is a list of file types that may be input. 659 is the first output substitute object information item, and 663 is the nth output substitute object information item. Each output substitute object comprises its own object ID 660, a corresponding data output object ID that indicates the ID of the corresponding data output object and a compatible file type list 662, which is a list of file types that may be output.

[0059]

The operation of the present invention will now be explained with regard to the control system for multimedia devices 2 based on the system control method described above, using a digital VTR as a specific example.

[0060]

Fig. 14 is a drawing showing the situation before the object-rendered digital VTR is connected to the multimedia controller. In Fig. 14, 203 is a digital VTR, and 206 is a digital VTR object that resides in the digital VTR 203 at all times and functions as an object-rendered digital VTR as seen from other devices on the LAN. The digital VTR object 206 further comprises three objects. 207 is a digital VTR controller object that controls the hardware of the digital VTR 203.

[0061]

208 is a digital VTR data input object that is responsible for the input of digital data from other devices via the LAN 4. 209 is a digital VTR data output object that is responsible for the output of digital data to other devices via the LAN 4. 210 is a digital VTR substitute object description file that describes the specifications of the digital VTR substitute object that is generated inside the multimedia controller 1 when the digital VTR 203 is connected to the multimedia controller

1 via the LAN 4.

[0062]

The digital VTR substitute object description file 210 comprises a digital VTR control panel object description 211 that describes the specifications of the operation panel for the digital VTR 203 and a digital VTR data I/O substitute object description 212 that describes the specifications of the digital VTR data I/O substitute object that performs the substitution in the data I/O regarding the digital VTR 203.

[0063]

Fig. 15 is a drawing showing the construction of the VTR controller object 207. In the drawing, 1009 is a class method table pointer storage unit, and stores the pointer to the class method table 1018. The class method table 1018 comprises a number of data processing means, including the play execution means 1019 that controls the hardware of the digital VTR 203 and executes the play-back operation, and the recording execution means 1020 that executes the recording operation. 1010 is a message communication means and 1011 is a process search means. 1012 is a method unit, but the actual data processing means is indicated by means of the class method table 1018. 1015 is an internal data unit and comprises a number of variables and status information items necessary for the control of the digital VTR 203, including the tape movement status 1016 and the tape current position 1017.

[0064]

First, the operation when the digital VTR 203 is connected to the LAN 4 will be explained. Fig. 16 is a drawing showing the sequence of steps when the digital VTR 203 is connected to the LAN 4. Fig. 17 is a drawing showing the screen of the multimedia controller 1. In Fig. 17, 228 is a display of the multimedia controller 1, 229 is an icon display indicating that the digital VTR 203 is connected, and 230 is a cursor that

indicates where the pointing device, such as a mouse, is pointing. Although the pointing device is not shown in the drawings, it is equipped with a button. The operation by which the user presses the button and then immediately releases it is generally called 'clicking', and the operation in which clicking is performed twice within a certain interval is called 'double-clicking'. For other connectable devices, various devices such as a camera (still image input), a tuner, a television, various databases and a CD, may be connected. Selection and control of these devices may also be performed by means of the icon display on the screen 228.

[0065]

Fig. 18 is a drawing to explain the situation in which an object-rendered digital VTR 203, an example of the multimedia device, is connected to the LAN 4. In Fig. 18, 220 is an object generated inside the multimedia controller 1, and comprises a digital VTR substitute object 220 that functions as a substitute for the digital VTR 203 inside the multimedia controller 1. The digital VTR substitute object 220 consists of a digital VTR control panel object 221 that functions as a control panel for the digital VTR 203, a digital VTR data input substitute object 222 that functions as a substitute for the data input object 208 during data input, and a digital VTR data output substitute object 223 that similarly functions as a substitute for the data output object 209.

[0066]

The operation sequence when the object-rendered digital VTR 203, an example of the multimedia device, is connected to the LAN 4 will be explained with reference to Figs. 16, 17 and 18. When the digital VTR 203 is connected to the LAN (636), the system director object 205 recognizes the connection of the digital VTR 203 (637). The system director object 205 then sends a device ID to the digital VTR 203 (638).

[0067]

The system director object 205 loads the digital VTR substitute object description file 210 from the digital VTR 203 using the multimedia device substitute object generating means 1047 (639). The system director object 205 then generates the digital VTR substitute object 220 inside the multimedia controller 1 based on the digital VTR substitute object description file 210 using the multimedia device substitute object generating means 1047 (640). The connection status shown in Fig. 18 then results. The digital VTR substitute object 220 then displays the icon display 229 for the digital VTR 203 on the display 228 of the multimedia controller 1 (641). It then waits for an instruction from the user (642).

[0068]

The user now can control the digital VTR via the digital VTR substitute object 220 inside the multimedia controller 1 by operating the digital VTR based on the operation screen displayed based on the digital VTR control panel object 221 of the multimedia controller.

[0069]

The relationship between the descriptions of the digital VTR substitute object description file 210 and the generated objects will now be explained in more detail.

[0070]

Fig. 19 shows an example of the icon for the digital VTR 203, and Fig. 20 shows an example of the control panel display screen. Fig. 19 shows an icon 229 that is displayed when the digital VTR 203 is connected to the LAN 4. Fig. 20 is a default display screen drawn by the digital VTR control panel object 221. In this drawing, 232 is a display selection menu of the control panel displayed on the display, 265 is a time counter display that displays the time elapsed on the tape, 266 is a control mode selection unit by which to select the control mode

of the digital VTR 203, 267 is a first switch button display to set the default control mode, 268 is a second switch button display to select a more detailed control mode, 269 is a rewind button display, 270 is a REVERSE PLAY button display, 271 is a pause button display, 272 is a PLAY button display, 273 is a fast forward button display, 274 is a stop button display and 275 is a record button display.

[0071]

Fig. 21 is a drawing to explain the correspondence between the class to which each object belongs and the components of the digital VTR control panel object 221. The class to which each basic component belongs is pre-defined in the class library 1081 and is maintained inside the multimedia controller 1. As shown in Fig. 21, each component of the digital VTR control panel object 221 individually functions as an object comprising a part of the digital VTR control panel object 221.

[0072]

In Fig. 21, the frame of the control panel display screen 231 corresponds to the VTR control panel object 284 of the panel class (ID = 1). The display selection menu 232 of the control panel corresponds to the panel view setting menu object 285 of the menu class (ID = 2). The time counter display 265 corresponds to the time counter object 286 of the form class (ID = 3). The rewind button display 269 corresponds to the rewind button object 287 of the button class (ID = 4). The REVERSE PLAY button display 270 corresponds to the REVERSE PLAY button object 288 of the button class (ID = 5). The pause button display 271 corresponds to the pause button object 289 of the button class (ID = 6). The PLAY button display 272 corresponds to the PLAY button object 290 of the button class (ID = 7). The fast forward button display 273 corresponds to the fast forward button object 291 of the button class (ID = 8). The stop button display 274 corresponds to the stop button object 292 of the button class

(ID = 9). The record button display 275 corresponds to the record button object 293 of the button class (ID = 10).

[0073]

The control mode selection unit 266 corresponds to the control mode switching object 294 of the button group class (ID = 11). The first switch button 267 corresponds to the default button object 295 of the radio button class (ID = 12). The second switch button 268 corresponds to the superior button object 296 of the radio button class (ID = 13).

[0074]

Among the objects comprising the digital VTR control panel object 221 shown in Fig. 21, the generation of the PLAY button object PLAY button object 290 will now be explained as an example.

[0075]

Fig. 22 is a drawing to explain the generation of the PLAY button object 290. In Fig. 22, 297, 298, 299, 300, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610 and 611 indicate the components described in the object control panel object description 247 of the digital VTR substitute object description file 210.

[0076]

297 is an object recognition information item, and comprises a class name 298, an object ID 299 and an upper object ID 300. 601 is a first object draw information item, and comprises a draw position/size information item 602, a shape/color information item 603 and an object image 604. 605 is a second object draw information item, and comprises a draw position/size information item 606, a shape/color information item 607 and an object image 608. 609 is an object link information item, and comprises a link destination object ID 610 and a transfer message 611.

[0077]



290 is a PLAY button object, a button object that is generated from the class information and the information of the object control panel object description 247 of the digital VTR substitute object description file 210. 613 is a class method table pointer storage unit and stores the pointer that points to the button class class method table 625. The button class class method table comprises a button initializing means 626 that initialize the internal variables of the button object when a button class object is generated, a button draw means that draws the button object display, and a click response means that, when the user indicates the draw position of the button object by means of the cursor 230 for a pointing device such as a mouse and clicks on it, displays that the button object has been clicked by temporarily changing the display of the button in response to the clicking operation and sends a message to other objects.

[0078]

The definition regarding each data processing means held by the button class method table is described in the class and may be referred to and used not only by the PLAY button object 290 but also by all other objects that belong to the button class. 614 is a message communication means and 615 is a process search means. 616 is a method unit and 620 is an internal data unit. The internal data unit 620 comprises an object ID 621, a button status data 622, draw parameters 623 and link data 624. The types of internal data that should be held not only by the PLAY button object 290 but also by all button objects belonging to the button class are described in the class.

[0079]

The system director object 205 reads the digital VTR substitute object description file 210 and generates each object. In the example of Fig. 22, button class objects are generated based on the description of the class name 298 of the

object recognition information item 297. When the PLAY button object 290 is generated by the system director object 205, the internal data unit 620 is initialized by the button initializing means 626. In the example of Fig. 22, the object ID is set to be ID = 7 based on the description of the object ID 299. From the description of the upper object ID 300, the system director object 205 learns that the PLAY button object 290 belongs to the digital VTR control panel object 221. Based on the upper object information regarding each object, the system director object 205 learns the relationships among the objects and generates objects comprising multiple objects as compound objects.

[0080]

The button draw means 627 draws the PLAY button object 290 based on the draw parameters 623 and the button status data 622. The button draw means 627 is automatically executed when a button object is generated and when the upper object is moved.

[0081]

The first object draw information item 601 describes the draw information 625 for the button when it is not pressed. The draw position/size information item 602 describes the rectangular frame information that indicates the draw position and size of the PLAY button object 290 when it is drawn in the digital VTR control panel object 221. The rectangular frame information is expressed by means of coordinate information that defines the rectangular information in the coordinate system of the digital VTR control panel object 221, as in (X1, Y1), (X2, Y2) for the draw information 625 for the unpressed button, i.e., the left upper coordinates and the right lower coordinates, as an example. The drawing of the PLAY button object when it is not pressed is performed based on the shape/color information item 603 or the object image 604. The shape/color information item 603 is described using the

language of how to draw the object, such as how to draw lines and how to apply colors. The object image 604 is expressed by means of bitmap data. In general, while expression by means of the former requires a smaller amount of data, the latter has a larger degree of freedom.

[0082]

The second object draw information item 605 describes the draw information 626 for the button when it is pressed using the same method as the first object draw information item 601. The draw parameters 623 are determined based on the first object draw information item 601 and the second object draw information item 605. The link data 624 is set based on the object link information item 609, 'play' is set as a transfer message, and a link destination object ID is set as the link destination object ID. For the purpose of determining one receiving object for the entire system when a message is sent, the link data 624 is set in the form of the link destination object ID to which the device ID is added that was allocated to the digital VTR by the system director object 205 when the digital VTR 203 was connected to the LAN 4.

[0083]

Therefore, even if an ID allocated to two or more objects is used between devices, the message may be correctly transmitted. The button status data holds the status[?] regarding whether or not the button is pressed.

[0084]

Fig. 23 is a drawing showing a flow chart of the operation executed when the user places the cursor 230 on the icon display 229 for the digital VTR 203 and double-clicks on it, and the flow chart for the operation executed when the control panel is operated.

[0085]

Fig. 24 is a drawing showing the display screen of the

multimedia controller 1 when the user double-clicks on the icon display 229 for the digital VTR 203. In Fig. 24, 231 is a default control panel display screen for the digital VTR 203 and 272 is a play button.

[0086]

Fig. 29 is a drawing showing the construction of the digital VTR control panel object of the panel class and its relationship to the object description information.

[0087]

In Fig. 29, 1401 is a class method table pointer storage unit, and points to the panel class class method table 1402. The panel class class method table comprises a panel initializing means 1403 that initializes the panel objects, a panel draw means 1404 that draws the panel, and a click response means 1405 that indicates the operation executed when the panel is double-clicked. 1406 is a message communication means, 1407 is a process search means and 1410 is an internal data unit. 1411 is an object ID, 1412 is panel status data and 1413 is draw parameters. The internal data unit 1410 is initialized in accordance with the description in the digital VTR substitute object description file 210, while the digital VTR control panel object description 211 in the digital VTR substitute object description file 210 comprises an object recognition information item 1414, a first object draw information item that indicates the icon image 1426 for the digital VTR 203, and a second object draw information item 1422 that indicates the frame 1427 of the digital VTR control panel. The object recognition information item 1414 comprises a class name 1415 (panel class), an object ID 1416 (ID = 1) and an upper object ID. The first object draw information item 1418 comprises a draw position/size information item 1419, a shape/color information item 1420 and an object image 1421. The second object image 1422 comprises a draw position/size information item 1423, a

shape/color information item 1424 and an object image 1425.  
[0088]

The method by which to instruct the display of the control panel and the play of the digital VTR 203 will be explained with reference to Figs. 23 and 29. At the time that the system director object 205 generates the digital VTR substitute object 220 in the operation explained with reference to Fig. 16, the digital VTR substitute object 220 displays the icon display 229 based on the icon image 1426. When the user indicates the digital VTR icon 229 by means of the cursor 230 and double-clicks on it (643), the control panel object 221, a component of the digital VTR substitute object 220, sends out a message instructing image draw to all of the objects that comprise the control panel object 221. All of the objects shown in Fig. 21 execute the draw means in accordance with this message. When this occurs, the control panel object draws the frame for the digital VTR control panel based on the second object draw information item. As a result, a digital VTR control panel display 231 to operate the digital VTR 203 is displayed, as shown in Fig. 24 (644), and waits for the instruction from the user (645). When this state is present and when the user indicates the play button 272 in the control panel 231 by means of the cursor 230 and clicks on it (646), the control panel object 221 sends out the message 'play' to the controller object 214 of the digital VTR 203 (647). Consequently, the controller object 214 of the digital VTR 203 activates the play execution means in response to this message (648). The play operation of the digital VTR 203 is begun due to the activation of the play execution means.

[0089]

As explained above, using the present invention, simply by connecting a multimedia device to the multimedia controller via a LAN, the multimedia device substitute object necessary to operate the multimedia device is automatically generated in

the multimedia controller, and further, the control panel necessary for the operation of the multimedia device is automatically displayed in the multimedia controller display, such that when the user operates the control panel, an appropriate message is sent to the controller object of the multimedia device and the desired operation may be performed. Because the information necessary to generate the multimedia device substitute object required for the operation of the multimedia device is obtained from the multimedia device substitute object description file that is read from the multimedia device, the multimedia controller need only have a basic class library and need not have information regarding any specific multimedia devices to begin with.

[0090]

Fig. 25 is a drawing showing the construction of the digital VTR data input substitute object and its relationship to the object description information. In Fig. 25, 222 is a digital VTR data input substitute object, and 668 is a class method table pointer storage unit that points to the data input substitute class class method table 679. The data input substitute class class method table 679 comprises a data input substitute object initializing means 680, a link information updating means 681 and a compatible file type response means 678.

[0091]

669 is a message communication means, 670 is a process search means and 671 is a method unit. 674 is an internal data unit, 675 is an object ID and 676 is a corresponding data input object ID, which is the ID of the corresponding data input object. 677 comprises compatible file types that indicate the data file types that may be input, and 1006 is a data output object link information item.

[0092]

The digital VTR data input substitute object is generated based on the digital VTR data I/O substitute object description 212 in the digital VTR substitute object description file 210. 682 is an input substitute object information item described in the digital VTR data I/O substitute object description 212 and comprises an object ID (ID = 1 in this example) 683, a corresponding data input object ID (ID = 1 in this example) 684, and a compatible file type list 685 (formats termed AV1 and AV2 in this example). The input substitute object initializing means 680 initializes the internal data unit 674 based on these descriptions.

[0093]

Fig. 26 is a drawing indicating the construction of the digital VTR data output substitute object and its relationship to the object description information. In Fig. 26, 223 is a digital VTR data output substitute object, and 690 is a class method table pointer storage unit that points to the data output substitute class class method table 1048. The data output substitute class class method table 1048 comprises a data output substitute object initializing means 694, a link information updating means 695 and compatible file type response means 700.

[0094]

691 is a message communication means, 692 is a process search means and 693 is a method unit. 696 is an internal data unit, 697 is an object ID and 698 is a corresponding data output object ID, which is an ID of the corresponding data output object. 699 is compatible file types indicating the file types of data that may be output, and 688 is a link information item regarding the data output object.

[0095]

The digital VTR data output substitute object is generated based on the digital VTR data I/O substitute object description 212 in the digital VTR substitute object description file 210.

1001 is a data output substitute object information item described in the digital VTR data I/O substitute object description 212 and comprises an object ID (ID = 1 in this example) 1002, a corresponding data output object ID (ID = 1 in this example) 1003, and a compatible file type list 1004 (formats termed AV1 and AV2 in this example). The data output substitute object initializing means 694 initializes the data in the internal data unit 696 based on these descriptions.

[0096]

Fig. 27 is a drawing showing the construction of the digital VTR data input object. In Fig. 22, 208 is a digital VTR data input object, and 1022 is a class method table pointer storage unit that points to the data input class class method table 1031. The data input class class method table 1031 comprises a file writing means 1032, a data receiving means 1033 and a link information updating means 686. 1023 is a message communication means, 1024 is a process search means and 1025 is a method unit. 1028 is an internal data unit, 1029 is an object ID and 1030 is a link information item.

[0097]

Fig. 28 is a drawing showing the construction of the digital VTR data output object. In Fig. 28, 209 is a digital VTR data output object, and 1035 is a class method table pointer storage unit that points to the data output class class method table 1044. The data output class class method table 1044 comprises a file reading means 1045, a data sending means 1046 and a link information updating means 687. 1036 is a message communication means, 1037 is a process search means and 1038 is a method unit. 1041 is an internal data unit, 1042 is an object ID and 1043 is a link information item.

[0098]

When the data input substitute object 222 and the data output substitute object 223 for the digital VTR 203 are

generated inside the multimedia controller, they function as if they were the digital VTR data input object 208 and the digital VTR data output object 209. For example, where a file of another multimedia device is to be copied to the digital VTR, the system director object 205 asks for the file types that may be input to the digital VTR data input substitute object 222. In response to the inquiry from the system director object 205, the compatible file type response means of the digital VTR data input substitute object 222 responds with the file types that may be received by the digital VTR 203.

[0099]

Where the file type of the file to be copied is among them, a link from the output substitute object of the multimedia device on which the file to be copied resides to the digital VTR data input substitute object 222 is established. The link information updating means 681 of the digital VTR data input substitute object 222 sends a message to the digital VTR data input object 208 to activate the link updating means of the digital VTR data input object 208 and updates the link information 688 of the digital VTR data input object 208.

[0100]

At the same time, the data output substitute object of the multimedia device on which the file to be copied resides sends a message to update the link information of the data output object and updates the link information, through which a link is established between the data output object of the multimedia device on which the file to be copied resides and the digital VTR data input object 208.

[0101]

The data sending means of the data output object of the multimedia device on which the file to be copied resides is activated, and the data output object of the multimedia device on which the file to be copied resides sends a message to the

digital VTR data input object to activate the data receiving means 1033 and the file writing means 1032, as a result of which the file is copied. In other words, when a copying instruction is supplied to the data input substitute object and the data output substitute object inside the multimedia controller, the data input substitute object and the data output substitute object each send a message to the data input object and the data output object of the multimedia device and establish a data link between the multimedia devices. Consequently, it is not necessary for the multimedia controller to be directly involved in the actual data copying, etc.

[0102]

As described above, using the present invention, when the entire system in which multiple multimedia devices are connected is controlled, it is no longer necessary to make such preparations as pre-installing in the controller device drivers or application software to control the multimedia devices, and by connecting a multimedia device to the LAN, its control panel and the device status are displayed on the screen of the controller, such that the power can be turned ON and OFF, the main unit can be controlled and input and output switching may be performed on the screen in an easier manner.

[0103]

In addition, among the components of the control panel that the multimedia device sent to the controller, those that share the same definition as those already possessed by the controller may be exchanged if the user so desires, such that user interfaces that vary from one manufacturer to another may be unified.

[0104]

Further, control from a remote controller and access to the multimedia devices may be performed transparently through a LAN.

[0105]

A more specific explanation will now be provided regarding the messaging within a system when the user copies data from a digital camera to a digital VTR in connection with a system in which multimedia devices and a controller such as those described above are connected via a LAN. The digital camera of this embodiment is a portable camera and can record moving images and sounds outdoors. It also has a means to communicate with said system.

[0106]

Fig. 30(a) is a control panel displayed on the multimedia controller display. 301 is a control panel to control the digital camera device, and 310 is a control panel to control the digital device. Each control panel is displayed on the multimedia controller display when the multimedia device control panel object description described above (1062 in Fig. 7) is read into the multimedia controller from each device by means of the system director, and waits for events from the user.

[0107]

In Fig. 30(a), 302 and 311 are tape insertion displays that show whether or not a tape is inserted in each device. 303 and 312 are counters that display the time elapsed on the tape inserted in each device. 304 and 313 are button objects to display a list of the contents of the tape inserted in each device, and by clicking on these button objects by means of the mouse, a list of the contents of the tape is displayed. 305 and 314 are PLAY button objects, and by clicking on these button objects by means of the mouse, each device executes the play function.

[0108]

306 and 315 are FF button objects, and by clicking on these button objects by means of the mouse, each device fast-forwards the tape. 307 and 316 are RWD buttons and by clicking on these button objects by means of the mouse, each device rewinds the

tape. 308 and 317 are stop buttons, and by clicking on these button objects by means of the mouse, each device stops the play, FF, RWD or REC function. 309 and 318 are REC button objects, and by clicking on these button objects by means of the mouse, each device starts recording of image and sound data externally input via a communication medium such as a LAN.

[0109]

The procedure when image and sound data are copied from the digital camera 301 to the digital VTR 310 through the operation of such a user interface will now be explained below with reference to Fig. 30.

[0110]

When the user begins a dragging operation from inside the control panel 301 in the user interface shown in Fig. 30(a), the black frame 319 is displayed as shown in Fig. 30(b). If dragging is continued on the display, an arrow 321 is displayed when the cursor leaves the black frame 319 and the tip of the arrow indicates the position of the mouse cursor. If the mouse continues to be moved, when the mouse cursor enters the frame 310, a black frame 320 is displayed. If the dragging is finished there, a between-object link is established from 301 to 310.

[0111]

At this point, determination of the validity of the link begins inside the multimedia controller. If the link is invalid, the display returns to the status shown in Fig. 30(a), and if the link is valid, the display is maintained in the status shown in Fig. 30(b).

[0112]

The processes inside the multimedia controller where a link is established based on the user's operation will be explained below with reference to Figs. 31, 32 and 33.

[0113]

Fig. 31 shows the construction of the objects inside the

multimedia controller 322, the VTR device 338 and the digital camera device 339 of this embodiment and the connection status among the devices. All of the objects inside each device may mutually send and receive messages and data. In addition, they can send and receive messages to and from objects residing in other devices via the LAN by means of their communication means 336, 406 and 407. Therefore, all of the objects shown in Fig. 31 can send and receive messages and data to and from any other object.

[0114]

The relationships among the objects shown in Fig. 31 are expressed by the upper object ID in the basic construction of the object. 323 is the system director object. The system director object 323 has in its method unit a means (data I/O managing means) 324 to determine the compatibility of data I/O when data is input and output between devices. The connected device substitute object storage unit 335 stores the substitute object of each device that is generated by reading from each device connected to the LAN 341 the prescribed information necessary for initialization at system startup as well as at certain intervals by means of the connected device substitute object initializing means.

[0115]

In this embodiment, the digital VTR substitute object 326, the digital camera substitute object 331 and other device objects not shown in Fig. 31 are generated by reading the initialization information from the digital VTR device 338, the digital camera device 339 and other devices not shown in Fig. 31 that are connected to the LAN 337. 325 is a digital VTR data input substitute object, and has a means to respond to the inquiries regarding data input to the digital VTR device. The digital camera data output substitute object 330 has a means to respond to the inquiries regarding data output from the

digital camera device. The digital VTR device controller object 340 controls the hardware of the digital VTR device in response to the messages from other objects. The digital camera device controller object 341 controls the hardware of the digital camera device in response to the messages from other objects. The digital VTR device input object 408 has a data receiving means and can receive data that is sent to its object ID. The digital camera output object 409 has a data sending means and sends data by means of a communication means 407 by adding information such as a data destination ID to the data reproduced by the play device of the digital camera device not shown in the drawings. The communication means 336, 406 and 407 are communication means to perform communication with the devices connected to the LAN.

[0116]

Fig. 11 shows the construction of the system director object. When a message indicating that a link has been established between devices based on the user's input is received from the window server, the process search means 342 activates the data I/O managing means 343. The data I/O managing means 343 performs a series of processes to determine the data compatibility between the devices for which the user has established a link.

[0117]

Fig. 32 is a flow chart showing the sequence of steps followed by the data I/O managing means. When the user establishes a link from the device A to the device B by means of the user interface as shown in Fig. 30, the window server gives the system director object (205 in Fig. 11) a message that a link has been established from the device A to the device B (LINKED message). The process search means (342 in Fig. 11) that has received this message activates the data I/O managing means from the method unit.

[0118]

The sequence of steps followed by the data I/O managing means when a link is established from the device A to the device B by the user will be explained below with reference to the flow chart of Fig. 32.

[0119]

First, in S1, the data I/O managing means asks the output substitute object of the device A for its compatible file types (where there are multiple compatible file types, a list of them is returned).

[0120]

In S2, the data I/O managing means asks the input substitute object of the device B for its compatible file types. Where the device B has multiple compatible file types (compatible formats), the input substitute object of the device B gives the information that indicates the order of priority among the file types at the same time. Here, the order of priority means the order of the file types desirable for input by the device B as designated by the user or the device B manufacturer. The file type at the top of the order of priority is termed the first priority file type for the device B. In S3, the data I/O managing means searches the compatible file types (or the compatible file type list) of the device A in accordance with the order of priority for the device B. In other words, the first priority file type for the device B is sought from among the compatible file types of the device A read in S1. The file type that was obtained in this process is termed the most compatible file type for the devices A and B.

[0121]

In S4, where the search in S3 failed (i.e., when there was no matching file type in the compatible file type list for the device A and the compatible file type list for the device B), the data I/O managing means advances to S8, and where the search

succeeded, it advances to S5.

[0122]

In S5, the data I/O managing means determines whether or not the data attribute of the most compatible file type obtained in S3 matches the data attribute of the first priority file type for the device B, and if they match, the process of S6 is executed.

[0123]

In S6, the link is determined to be valid, and the fact that a valid link has been established from the device A to the device B, the data attribute regarding the link and the file type are stored in the inter-device link information management data (344 in Fig. 11) residing in the system director object internal data unit.

[0124]

In S7, the fact that a valid link has been established from the device A to the device B and the file type regarding the link are sent to the LINKED message sender object, the output substitute object of the device A and the input substitute object of the device B (the notification to the input substitute object and the output object of each device. [THE MEANING OF THE ORIGINAL IS UNCLEAR.]), whereupon the routine comes to an end.

[0125]

Where the compatible file types do not match in S4 and the data I/O managing means advances to S8, the link is determined to be invalid and the LINKED message sender object is notified that the link is invalid due to the lack of a common format. In S11, a message requesting the cancellation of the link is sent to the LINKED message sender object, whereupon the routine comes to an end.

[0126]

Where the data attribute of the most compatible file type and that of the first priority file type do not match in S5,



a message requesting a warning display is sent to the LINKED message sender object. The contents of the warning comprise a question to the user asking whether the user will be satisfied with a data transfer using the data attribute of the most compatible file type, and leads to an event loop in which the data I/O managing means waits for input by the user regarding whether to continue or cancel the link. Where the user instructs that the link be continued in S10, the process of S6 is executed, and where the user instructs that the link be canceled, the process of S11 is executed, and after a link cancellation request message is sent, the routine comes to an end.

[0127]

The sequence of steps within the multimedia controller when the user has established a link from the digital camera control panel (301 in Fig. 30(a)) to the digital VTR control panel (310 in Fig. 30(a)) will be explained more specifically below with reference to Figs. 30, 11 and 33.

[0128]

When the user establishes a link 321 from the digital camera control panel to the digital VTR control panel as shown in Fig. 30(b) on the user interface shown in Fig. 30, the window server sends to the system director object in Fig. 11 a message that a link has been established from the digital camera control panel to the digital VTR control panel.

[0129]

Upon receiving this message, the process search means 342 of the system director object activates the data I/O managing means 343 from the method unit. The data I/O managing means 343 follows the routine shown in the flow chart of Fig. 33 and determines whether the link established between the digital camera control panel and the digital VTR control panel is valid or invalid.

[0130]

First, in S1, the data I/O managing means 343 asks the digital camera output substitute object for its compatible file types. The table shown in Fig. 45(a) shows the compatible file types for the digital camera of this embodiment. Since it has multiple compatible file types, they are shown in the form of a list. The compatible file types are shown together with the data attributes, as shown in Table 1. Here, Audio indicates a sound data attribute, Movie indicates a moving image data attribute and AudioMovie indicates a synchronized sound and moving image data attribute.

[0131]

In S2, the data I/O managing means 343 asks the input substitute object of the digital VTR for its compatible file types. The digital VTR of this embodiment has multiple compatible file types, as shown in the table in Fig. 45(b), and each file type is given together with its order of priority. In this embodiment, as shown in Table 2, the file type AM4 having an AudioMovie data attribute is the first priority file type for the VTR device.

[0132]

In S3, the data I/O managing means 343 searches for the matching file type from the table in Fig. 45(a) in accordance with the order of priority in the table in Fig. 45(b). Because AM4 has the first priority based on the table in Fig. 45(b), the data I/O managing means 343 first searches for the AM4 format from the table in Fig. 45(a). Since the table in Fig. 45(a) does not have the AM4 format, it then searches for AM5, which has the second priority, from the table in Fig. 45(a).

[0133]

If this search also fails, the data I/O managing means 343 continues searching for the matching file type in the same manner in accordance with the order of priority. A matching file type is found when it searches for Movie2, which has the fourth

priority, and this is deemed the most compatible file type for the link from the digital camera to the digital VTR.

[0134]

In S4, since a matching file was found during the search in S3, the data I/O managing means 343 advances to S5.

[0135]

In S5, because the data attribute Movie of Movie2, which is the most compatible file type obtained in S3, does not match the data attribute AudioMovie of the first priority file type of the digital VTR, the data I/O managing means 343 advances to S8.

[0136]

In S8, the data I/O managing means 343 sends the window server a message requesting a warning display as shown in Fig. 34 and enters the user input wait loop S12.

[0137]

In this embodiment, it is assumed that the user has clicked on YES from the panel in Fig. 34 and the data I/O managing means 343 advances from S10 to S6.

[0138]

In S6, the fact that a valid link for the file format Movie2 having the data attribute Movie has been established from the digital camera to the digital VTR is stored in the inter-device link information management data (344 in Fig. 34) of the system-director internal data unit.

[0139]

Finally, in S7, the data I/O managing means sends a message to the digital camera output substitute object and the digital VTR input substitute object and notifies them of the fact that a link for the data attribute Movie2 has been established from the digital camera to the digital VTR, whereupon the routine for the data I/O managing means 343 comes to an end.

[0140]

Upon receiving the message from the data I/O managing object, the digital camera output substitute object (410 in Fig. 41) activates the link information updating means (413 in Fig. 41) of the method unit. The activated link information updating means (413 in Fig. 41) stores in the link information item (419 in Fig. 41) of the internal data unit the fact that a link has been established to the digital VTR device for the Movie2 format and notifies the digital camera data output object (420 in Fig. 42), which is the corresponding data input object ID, that the link information has been updated, as well as the contents of the update. Upon receiving this notification, the digital camera data output object (420 in Fig. 42) activates the link information updating means (425 in Fig. 42) and stores the contents of the update in the link information item (428 in Fig. 42).

[0141]

The digital camera output substitute object (410 in Fig. 41) sends out a message to all the objects in the digital camera control panel object (333 in Fig. 31), ordering those objects not involved in the data output to become grayed out so that they will be disabled from receiving any input from the user. Based on this message, the REC button 309 in Fig. 30(b) is grayed out and does not accept mouse clicks from the user.

[0142]

The digital VTR input substitute object (222 in Fig. 25) that has received a message from the system director object activates the link information updating means 681. The activated link information updating means 681 stores the fact that a link has been established from the digital camera device for the Movie2 format in the link information item 1006 of the internal data unit and notifies the contents of the update of the link information to the digital VTR data input object (208 in Fig. 27), which is the corresponding data input object ID.

Upon receiving this notification, the digital VTR data input object (208 in Fig. 27) activates the link information updating means (686 in Fig. 27) and stores the contents of the link update in the link information item (1030 in Fig. 27). The digital VTR input substitute object (222 in Fig. 25) further sends a message to all the objects in the digital VTR control panel object (328 in Fig. 31), ordering those objects not involved in the data input to become grayed out so that they will be disabled from receiving any input from the user.

[0143]

Based on this message, the PLAY button 314, the FF button 315 and the RWD button 316 are grayed out and do not accept mouse clicks from the user, such that the existence of a valid link is maintained, as shown in Fig. 61(b).

[0144]

The status of the internal data when a valid link is present as shown in Fig. 30(b) is shown in Fig. 44. In Fig. 44, (a) is the digital camera output substitute object, (b) is a digital VTR input substitute object, (c) is the digital camera data output object, (d) is the digital VTR data input object and (e) is the internal data of the system director object. The internal variables that were updated due to the link established by the user are shown in brackets using italics.

[0145]

When a valid link is present as shown in Fig. 30(b) and the internal data is as shown in Fig. 44, if the user clicks on the PLAY button 305 307 [ORIGINAL MISTAKE], the digital camera 301 begins to transfer the data to the digital VTR in the Movie2 format.

[0146]

Further, when the user clicks on the REC button 318, the digital VTR 310 begins to record the data being sent to it from the digital camera 301 in the Movie2 format, whereupon copying

between the devices takes place. The messaging that takes place at this time inside the multimedia controller and between the devices will be explained below.

[0147]

First, when the user clicks on the PLAY button (305 in Fig. 30), the window server notifies the control panel object (333 in Fig. 32) of the coordinates of the point that was clicked on (the coordinates inside the control panel). Upon receiving this notification, the control panel object refers to its internal data and learns that the PLAY button is displayed at the point of these coordinates. It determines accordingly that the PLAY button has been clicked on and notifies the digital camera device controller object 341 that a data transfer order (PLAY message) has been issued by the user. The process search means 431 of the digital camera controller object (429 in Fig. 43) activates from the method unit the play execution means 433 that corresponds to this message. The play execution means 433 controls the hardware of the digital camera device to enable the mechanical parts of the digital camera device to begin playback immediately, while it sends to the digital camera data output object (409 in Fig. 31) an output request message for the data being played. Upon receiving this message, the process search means (421 in Fig. 42) of the digital camera data output object activates the data reading means 423 and the data sending means 424 from the method unit. The activated data reading means then reads the information from the recording medium.

[0148]

The activated data sending means first refers to the link information item 428. When this takes place, the contents of the link information item 428 comprises the data destination object ID = 120 and the file type = Movie2 as shown in 430 in Fig. 44. Therefore, the data sending means 424 converts the read data into the Movie2 file type and sends the data to ID = 120.

In the embodiment, Movie2 data comprises moving image data and has data required for the moving image playback, such as the file type and the numbers of pixels in the horizontal and vertical lines, as a header at the top area of the data as shown in Fig. 36. The main part is subdivided by means of time stamps indicating the time information, such that editing may be performed by the time stamp.

[0149]

Upon detecting that Movie2 data has been sent to it from the object having an ID = 110, the digital VTR device data input object (208 in Fig. 27) having an ID = 120 refers to its link information item (1030 in Fig. 27). Upon confirming that there is a link for the Movie2 file type between itself and the data output object having an ID = 110, as shown in 431 in Fig. 44, the digital VTR device data input object stores the header of the data that is being sent to it in the memory (370 in Fig. 37).

[0150]

If the user clicks on the REC button in the digital VTR control panel (318 in Fig. 30) at any point in time, the window server notifies the control panel object (328 in Fig. 31) of the coordinates of the point that was clicked on (the coordinates within the control panel). Upon receiving this notification, the control panel object refers to its own internal data and learns that the REC button is displayed at these coordinates. After determining accordingly that the REC button has been clicked on, the control panel object then notifies the digital VTR device controller object 341 that a data recording order (REC message) has been issued from the user. Upon receiving this notification, the process search means of the digital VTR controller object (1011 in Fig. 15) activates the recording execution means (1020 in Fig. 15) from the method unit. The activated recording execution means (1020 in Fig. 15)

controls the hardware of the digital VTR device to enable the mechanical parts of the digital VTR device to begin recording immediately and then sends the digital VTR data input object (208 in Fig. 27) a message requesting recording of the data being input. The digital VTR data input object (208 in Fig. 27) that has just been asked to record the data controls the hardware of the digital VTR device to begin recording onto the magnetic recording medium 374 by reading the header information (364 in Fig. 36) stored in the RAM (370 in Fig. 37) and splicing the data on and after the time stamp that is sent next after the header information. The Movie2 moving image data recorded in this way contains header information that is needed for the playback of moving images, and can therefore be played.

[0151]

<Second Embodiment> The second embodiment of the present invention will be explained below. The multimedia controller of this embodiment has the characteristics of the first embodiment and also has an application software (utility software) operating environment.

[0152]

The multimedia controller of this embodiment will be explained below. (In the present invention, application software and utility software are essentially identical, and therefore the term 'application software' will include utility software as well.)

[0153]

Fig. 39 shows the construction of the objects inside the multimedia controller of the second embodiment of the present invention. The multimedia controller of the second embodiment of the present invention has an installation means and an operating environment for application software (utility software) for the multimedia controller [devices?]. (In the present invention, application software and utility software

are essentially identical, and therefore the term 'application software' will include utility software as well.)

[0154]

In the multimedia controller of this embodiment, an application software program is installed in the application class storage unit (388 in Fig. 39) inside the multimedia controller from a floppy disk or a communication means. The application software program stored in the application class storage unit constitutes an object-oriented class description file, and the application object generating means (386 in Fig. 39) of the system director generates executable objects of the application software program in the application object operating area (389 in Fig. 39) by using this class, such that the application software program becomes ready for execution. An operation by the user using a pointing means such as a mouse regarding a window object comprising an application object drawn and made visible on the multimedia controller display is communicated to the application object as a message indicating the operation performed and the corresponding coordinate point on the window. Because the application object possesses within its own internal data information regarding the graphical images present at any given coordinate point on the window, it can recognize the clicking on icons and dragging between icons on the application window using the information from the window server.

[0155]

The connection constructor application will be explained below as a specific example of a multimedia controller application software program.

[0156]

Fig. 38 shows the user interface of the connection constructor. In Fig. 38, 370 is a connection constructor window, 379 is a mouse cursor, which is a user's input means, 371 through

378 and 383 are icons for a CD player, a VTR, a display, a CATV decoder, a DAT deck, an amplifier that amplifies the sound signals and that generates sounds from speakers not shown in the drawings, a magneto-optical player that performs playback and recording of a magneto-optical disk, the multimedia controller and an LD player, respectively. The data transfer relationships (connections) among the devices are indicated by means of arrows, such that the user can easily learn the data transfer relationships among the devices based on these arrows.

[0157]

In this embodiment, no connection has been established with regard to the LD player (383 in Fig. 38). If the LD player is made to perform playback in this situation, its image and sound data is sent to the objects inside the multimedia controller, which is the default link destination, that govern the image display and sound output.

[0158]

The connection constructor window display explained above can allow not only display of the connections but also editing of the connections on the window. The user can establish a connection by performing dragging between icons. Fig. 38 shows the situation in which the user is trying to establish a connection from the CD player 371 to the amplifier 376 by dragging the mouse.

[0159]

When the user clicks on the connection constructor icon (connection constructor object) displayed on the multimedia controller display (27 in Fig. 4), the window server sends the connection constructor object (390 in Fig. 39) a message that the icon has been clicked on. Upon receiving this message, the connection constructor object activates the connection constructor window display means (391 in Fig. 39) by means of the process search means, which is not shown in Fig. 39.

[0160]

The messaging that takes place among the objects inside the multimedia controller when the user clicks on the connection constructor icon and that pertains mainly to the connection constructor object 364 will be explained below with reference to the flow chart of Fig. 40.

[0161]

First in S1, the connection constructor window display means (391 in Fig. 39) requests from the system director object (205 in Fig. 11) a list of object IDs for the devices connected to the network. In the step of S2, it then waits until it receives the data corresponding to the message. When the system director object refers to the object registration information (1078 in Fig. 11) inside its internal data unit and returns the list of the registered device objects, this data is stored in the internal data unit and the connection constructor window display means advances to S3.

[0162]

In S3, the connection constructor window display means refers to its internal data unit and sends a message requesting icon graphical data from all the object IDs included in the list of connected device object IDs that is stored in the internal data unit. It waits for the data to be received in S4, and when it receives the graphical data from all the object IDs, the connection constructor object stores this data in its internal data unit and advances to S5. In S5, the connection constructor object sends the data I/O managing object a message requesting the sending of the inter-device link information, which clarifies what data attribute links are present between what object IDs. When the inter-device link information is received in S6, the connection constructor object now has information sufficient to display the connection constructor window as 370 in Fig. 38, and in S7, it draws the connection constructor window.

The image draw is performed after the icon display positions are calculated so as to ensure that linking lines do not cross to the extent possible.

[0163]

The links established between devices are expressed by means of straight lines connecting the icons representing the devices, as shown in Fig. 38, so that the user can easily determine the connection status. (However, the linking lines need not be straight. Curved lines may also be used in order to make the connection relationships easier to see.) In addition, the effective data attributes for each link are distinguished from one another by means of the type (or the color) of the line.

[0164]

In this embodiment, as shown in Fig. 38, the Visual data is displayed using a solid line, the Audio data is displayed using a dotted line, an application program is displayed using a chain line and text data is displayed using two-dot chain line.

[0165]

For example, in Fig. 38, the VTR device icon 372 and the display device icon 373 are connected using a solid arrow 379. This means that if playback is activated in the VTR device, the image data being played by the VTR device is automatically sent to the display device. If the user wants to watch CATV, simply by means of the user opening the CATV control panel and selecting a channel, the CATV image data is automatically sent to the ID of the display device 373 and the CATV sound data is automatically sent to the ID of the amplifier device 376.

[0166]

In this embodiment, data from the magneto-optical player is program data. When program data is sent to the multimedia controller, the multimedia controller detects the description 'program' for the data attribute in the header of the data and recognizes the data to be an application program, such that this

program is automatically launched. This function can be obtained not only in the multimedia controller but also in other devices. For example, if an operating system is installed in the display device and the display device receives a program, this program will be executed.

[0167]

In this embodiment, this program has image data embedded in it, and by executing this program, images are shown on the display. The benefit of this method is that because the image data is embedded in the program, attention need not be paid to the data format used between the devices.

[0168]

By creating a link between icons for the devices displayed on the window by means of the input means such as a mouse directly on the window, the user can edit the data transfer relationships among the devices. When a connection is established between icons, the connection editing means 394 sends a LINKED message to the system director object. Upon receiving the LINKED message, the system director object performs processing in accordance with the sequence shown in Fig. 33, as in the first embodiment. Where the link is determined to be valid by the data I/O managing means of the system director, the internal data regarding the objects involved in the link is updated in the same manner as in the first embodiment.

[0169]

The connection constructor object is notified by the system director object of the data attribute for the link established in S7 of Fig. 33. The connection constructor object selects the type of line to be used for the link in accordance with this data attribute and displays it. Therefore, the internal data regarding the objects involved in the link changes and the display on the connection constructor window match at all times.

[0170]

#### Effect of the Invention

As described above, using the system device control method of the present invention, when the entire system in which multiple multimedia devices are connected is controlled, it is no longer necessary to make such preparations in the controller as pre-installation of controller device drivers or application software to control the multimedia devices, and simply by connecting a multimedia device to the LAN, its control panel and the device status are automatically displayed on the screen of the controller, such that the power can be turned ON and OFF, the main unit can be controlled and input and output switching may be performed on the screen. Further, the user can establish data transfer relationships among the multimedia devices by means of a simple user interface. Moreover, because adjustments regarding the file format for data transfer are automatically made by the multimedia controller, the user need not perform the troublesome task of file format adjustment.